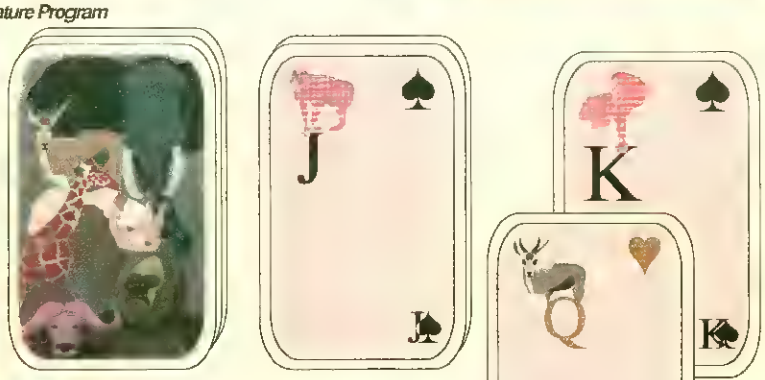
Serving the CoCo Community for

The RAINBOW 12 YEARS

THE COLOR COMPUTER MONTHLY MAGAZINE

January 1993 Vol. XII No. 6 Canada $4.95 U.S. $3.95

Feature Program



A Safari into Solitaire

by Kenneth Reighard, Jr.

We've all heard the phrase "all work and no play makes Jack a dull boy." Whatever the reason, it does sometimes help to play a bit with the computer so you don't feel overcome by work. Darn It, an addictive solitaire game written for the CoCo 3, is ideal for this purpose.

In Darn It, six columns of six cards each are displayed face up in the upper portion of the screen. The rest of the deck (16 cards) is placed face down in the lower-left area, and the top card of this deck is turned face up on the right. This latter card is the "play" pile.

The object of Darn It is to move all the cards from the top portion of the screen to the play pile. Legal moves are those in which the card to be played has a value exactly one higher or lower than the value of the face-up card at the top of the play pile. The suit of the cards is irrelevant. The card values go from Ace (low) to King (high). The values don't wrap around; you can't play a King on an Ace or an Ace on a King. In fact, you can't play any card on a King. You can move only the bottom cards from each column. To move, use the left and right arrows to select the column from which you want to play a card and press ENTER to move it to the play pile.

When none of the bottom cards in the column can be played, turn the next card on the deck face up on the play pile. To do this, simply press the space bar. The number of cards remaining in the deck is indicated on the back of the deck.

When the deck is empty and no legal moves are left, the game is over. The

See Safari on Page 10

Feature Program

OS-9: In From the Code

by Bruce Green

I have often found myself fumbling through my printer and OS-9 manuals looking for control codes to configure the printer or a window. Memorizing many codes seemed almost impossible and procedure files are too clumsy and limiting for this purpose.

Sendcode is a utility I wrote to eliminate the tedious chore of looking up these control codes. Instead, you use words to tell Sendcode what function you want, and it looks up the proper control codes to send.

Sendcode brings user-friendly operation to OS-9, especially with printers and other devices, by replacing numbers with simple names. When you need to send special control codes to a device, such as a printer, you probably dig out your manuals and use OS-9's `display` command to send the codes. For example, if you have a DMP-132 and want to turn on the underline function, you would refer to the printer manual to determine that the proper control code is $F. Then you would enter the following command to actually send the code:

```
display 0f > /p
```

Sendcode eliminates this hassle by using device-specific (one for each device you want to use) `.code` files in which you define simple command words for specific devices. The `.code` file is a simple list of names you want to use for device functions, along with the appropriate control-code sequences to perform these functions. Then you use this command word with Sendcode to actually perform the function. For the DMP-132 underline function above, you might enter something like

```
sendcode UndrlnON
```

Sendcode looks for the parameter `UndrlnON` in the `.code` file to determine the appropriate codes to send, then sends them. It's that easy.

To get started, you first need to enter and compile the `sendcode.c` source code shown in the listing. (Alternatively, the compiled program is on this month's RAINBOW ON DISK as well as in the OS9 Online SIG on DelphiÙ.)

After compiling the program, you need to create a `.code` file. Since most users will probably use Sendcode to send codes to a printer, I set the default `.code` file to

See Code on Page 17

In this issue:

THE RAINBOW

Editor and Publisher
Lawrence C. Falk

Managing Editor Cray Augsburg
Associate Editor Sue Fomby
Submission/Reviews Editor
Julie Hutchinson
Technical Editor Greg Law
Technical Assistant Ed Ellers
Contributing Editors Tim Kientzle
Matrin Goodman, M.D., Eddie Kuns
Consulting Editor Laura Falk

Art Director Debbee Diamond
Designers Sharon Adams, Heidi Nelson

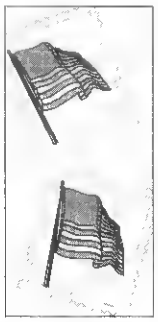
Director of Creative Services
O'Neil Arnold
Editorial Director John Crawley

Falsoft, Inc.

President Lawrence C. Falk
General Manager Judi Hutchinson
Asst. General Manager for Finance
Donna Shuck
Asst. General Manager for Administration
Tim Whelan
Administrative Asst. to the Publisher
Ellen Patterson
Bookkeeper/Dealer Accounts
Debra Wilson
Customer Service Manager
Bev Bearden
Customer Service Representative
Lauren Yates
Business Assistant Shannon Yoffe
Corporate Business Technical Director
Calvin Shields
Chief of Building Security
and Maintenance
Lawrence Johnson
Advertising Assistant Carolyn Fenwick
Western Sales Director Ira Barsky
(312) 587-1818
Eastern Sales Director Kim Lewis
(502) 228-4492

For RAINBOW Advertising and Marketing office information, see Page 19.

THE RAINBOW is published every month of the year by FALSOFT, Inc., The Falsoft Building, 9509 U.S. Highway 42, P.O. Box 385, Prospect, KY 40059, phone (502) 228-4492. THE RAINBOW, RAINBOWfest and THE RAINBOW and RAINBOWfest logotypes are registered trademarks of FALSOFT, Inc. ● Second class postage paid Prospect, KY and additional offices. USPS No. 706-650 (ISSN No. 0749-4777). POSTMASTER: Send address changes to THE RAINBOW, P.O. Box 385, Prospect, KY 40059. Authorized as second class postage paid from Hamilton, Ontario by Canada Post, Ottawa, Ontario, Canada. GST No. R125434092. ● Entire contents copyright by FALSOFT, Inc. 1992. THE RAINBOW is intended for the private use and pleasure of its subscribers and purchasers and reproduction by any means is prohibited. Use of information herein is for the single end use of purchasers and any other use is expressly prohibited. All programs herein are distributed in an "as is" basis, without warranty of any kind whatsoever. ● Tandy, Color BASIC, Extended Color BASIC and Program Pak are registered ® trademarks of the Tandy Corp. ● Subscriptions to THE RAINBOW are $24 per year in the United States. Canadian rates are U.S. $36. Surface mail to other countries is U.S. $58, air mail U.S. $103. All subscriptions begin with next available issue. ● Limited back issues are available. Please see Notice for issues that are in print and their costs. Payment accepted by VISA, MasterCard, American Express, cash, check or money order in U.S. currency only. Full refund after mailing of one issue. A refund of 10/12ths the subscription amount after two issues are mailed. No refund after mailing of three or more magazines.



LETTERS

The Right Stuff

Editor:

After opening my November 1992 issue of THE RAINBOW, I found myself spending more time than usual reading the articles. On reflection, I think the opinions of Tony Podraza and the experiences of John Donaldson caused me to take a second look. Let me complement THE RAINBOW for choosing these types of articles. I believe these are the kinds of authors needed to inspire some of the rest of us readers.

The blend of program listings, advertisements, product reviews, help with Delphi and the articles mentioned above was about right from my point of view. I do, however, suggest one minor change to alert the reader of the "Corrections" section. Since colors seem to he available for each page, the color red would stand out better than black for the heading. The size of the "Corrections" heading and the placement were good following the "Letters to the Editor."

Farrell Kenimer
2601 W. Corrine Drive
Phoenix, AZ 85029-2510

Thanks for the kind words, Farrell. We also think the two articles you mentioned are not only appropriate but needed, and we hope to provide similar articles in future issues.

We generally try to fit "Corrections" where there is room, then alert readers in the "Table of Contents." On the other hand, nothing says we can't try to find the extra room on a "red" page.

Needs a Disk Controller

Editor:

I need help. After collecting all my hardware and software for the CoCo 3, I have discovered I don't have an I/O [disk] controller! To complicate things, I understand it is nearly impossible to get one at this time. Will someone please help me find or build one?

C.J. Ryan
USCGC Gallatin (WHEC-721)
FPO, AE 09570-3908

Looking for EARS

Editor:

I'm working on a special setup and need help finding EARS, a voice-recognition system originally sold by Speech Systems. Since that company is no longer marketing products in this community, I'm hoping one of the "older" Color Computer users will have a copy of this hardware/software product he might be willing to part with.

Randy Aalderink
341 Lincoln Avenue
Holland, MI 49423-3662

Trying to Secure a Book

Editor:

For the past few months I have been searching for a book by the title *Security Projects for the TRS-80 Color Computer* and sold by Brown's Enterprises. When I wrote to Brown's, my letter was returned marked "No Forwarding Address." I've also checked through library networks, all to no avail. The book isn't even listed in the library's master index, *Books in Print*. How can I get a copy of this book?

Tim Perry
9724 Whispering Winds
Indianapolis, IN 46234

The most current address we have for Brown's Enterprises is 119 Skyline Dr. R.H., Granbury, TX 76048. If this lead doesn't pan out, perhaps another reader can offer some assistance.

Pico CAD Abandoned

Editor:

I regret to announce that I have abandoned the *Pico CAD* program project I started began three years ago. I delivered to my customers all the program modules except the plotter driver and, most importantly, the window interface. As delivered, *Pico CAD* remains the most powerful CAD software I have seen for the CoCo; but in the last two years I have been unable to take the time necessary to put the windowing interface on it. Because of its primitive condition, I am now offering a refund of up to 50 percent to any dissatisfied *Pico CAD* customers. Interested parties may write to me at the address below.

Paul Light
Gravity Studio
500 Rolling Hills Place, #209
Lancaster, TX 75146

We're sorry to hear you've decided to drop the project. Still, let us be the first to commend you for handling your decision in a professional and responsible manner.

Wants a New Keyboard and LEDs

Editor:

Is there a way to attach an IBM-type keyboard to my CoCo 3? Also, how can I add a power-on LED indicator to the CoCo 3 and the disk drive so I'll know when they're on?

James Ruth
128 Seymour Avenue
Newark, NJ 07108

Both Owl-Ware and Frank Hogg Labs at one time sold an adapter that allows you to connect a standard PC keyboard to the CoCo. It should be easy to add power indicators to the CoCo 3 and a disk drive. We've forwarded a copy of your letter to Marty Goodman; perhaps he'll provide the mechanics in a future issue.

An Addition to *Versabase*

Editor:

Pastor James Altom added a line to the *Versabase* series of database programs I wrote, and I think it really makes a handy addition to the programs. The new line, which follows, displays the current file as a reminder:

```
95 LOCATE 18,20:PRINT"THE CURREN
T FILE IS:  "; F$
```

This line can be added to all programs in the *Versabase* series. If you use all the programs, the easiest way to add it is to save it by itself as an ASCII file, then use MERGE to add it to each separate program.

David Polonsky
Tudor Court, Apt. 15
800 N. Broad Street
Elizabeth, NJ 07208

In Search of CoCo

Editor:

I'm looking for a spare CoCo. I tried to locate one at Radio Shack but of course had no luck. I have been a die-hard CoCo user for about eight years. Does anyone have suggestions for a good CoCo compromise?

Jacqueline Hutton
15210 Sherman Way, #34
Van Nuys, CA 91405

Renewed Hope

Editor:

I have been an avid RAINBOW reader and subscriber for many years, and I still have every issue I've purchased. After seeing Tandy discontinue the CoCo 3, and magazine ads for software and hardware disappear (not to mention the decreasing size of THE RAINBOW), I thought the CoCo was dead.

Now, with all the information available on the Hitachi 6309, and the advent of two fairly new CoCo resources, I see new life breathing into my Color Computer. I am finally going to get actively involved. I have been reluctant to submit programs because I want to send in more than one on a disk. Is that acceptable?

One more note: I've renewed my subscription for another year; I would like to see THE RAINBOW back to its old self. I am sure many others feel the same way. I, for one, would be willing to pay more than $31 a year to have a nice bulky magazine the size THE RAINBOW used to be.

Christian Miller
6079 Buerman Road
Sodus, NY 14551

You may submit more than one program/article on a single floppy, though we much prefer you send each submission on a separate disk. The submissions we receive are bagged separately for evaluation and multiple submissions on one disk increase the likelihood of problems when we're putting a submission into production.

THE RAINBOW welcomes letters to the editor. Mail should be addressed to: Letters to Rainbow, The Falsoft Building, 9509 U.S. Hwy 42, P.O. Box 385, Prospect, KY 40059. Letters should include the writer's full name and address. Letters may be edited for clarity or to conserve space.

Letters to the editor may also be sent to us through our Delphi CoCo SIG. From the CoCo SIG> prompt, enter RAI to get to the Rainbow Magazine Services area of the SIG. At the RAINBOW> prompt, enter LET to reach the LETTERS> prompt, then select Letters for Publication. Be sure to include your complete name and address.

OWL-WARE

Proven Technology

On the Razor's Edge of the Color Computer Frontier

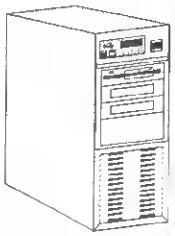
OWL Phones
Order Numbers (only)
1-800-245-6228
1-215-682-6855
Fax: 1-215-837-1942
Technical Help
1-215-837-1917

In our 10th Year !

A DECADE OF SERVICE TO THE CDMPUTER USER!

486SX-20 SYSTEMS - $1795.00!

Now You can enter the world of 4B6 computing at a reasonable cost!



the OWL SUPER ATOM - 486

High Powered Computing from a local, well established company.

- 33MHz / 50MHz 1486 based Systems with Socket for Weitek CoProcessor
- System and Video BIOS in Cache
- Large Tower Case : (33MHz, FCC Class B) - (50MHz, FCC Class A)
- 230 Watt Power Supply & 8 Option Slots
- System Price includes: 40MB HD, 4MB RAM, Std. Resolution Color VGA Monitor, High Resolution VGA Card, 2 High Density FD's, MS DOS 5.0

$1795 / $2095 / $2695 / $3595

486SX-20 ISA 486DX-33 ISA 486DX-50 ISA 486DX-50 EISA

- 105MB HD Upgrade Add $160.00
- Super VGA Upgrade Add $95.00

3- YEAR WARRANTY on all systems! Including One Full Year on Parts and Labor. Manufactures 3-Year Warranty on All Hard Drives

OWL SUPER ATOM - 386

$1565/$1645
25MHz 40MHz

- 25/40MHz 386DX Based
- Small Footprint Case
- FCC Class B Approved
- 200 Watt Power Supply
- 7 Expansion Slots
- 4MB of RAM
- 40MB Hard Drive
- Std. Resolution VGA Color Monitor
- 2 High Density FD's
- 101 Keyboard
- MS DOS 5.0

OWL SUPER ATOM - SX

$1295/$1349
1BMHz 25MHz

- 16/25MHz 386SX Based
- Small Footprint Case
- FCC Class B Approved
- 200 Watt Power Supply
- 7 Expansion Slots
- 2MB of RAM
- 40MB Hard Drive
- Std. Resolution VGA Color Monitor
- 2 High Density FD's
- 101 Keyboard
- MS DOS 5.0

386-SX Notebook Computers

20MHz, 60MB HD, 1.4MB FD, 2MB RAM(exp. to 5MB), VGA 640X480 LCD w/32 shades of gray. Ports: 2 Ser, 1 Par, 1 VGA, DOS & Windows, 7.7LBS!

$1545

386-DX Notebook Computers

33MHz, 120MB HD, 1.44MB FD, 32KB CACHE, 4MB RAM(exp. to 16MB), Std. VGA LCD w/32 gray, Ext. Keypad inc., DOS & Windows, 7.7LBS!

$1995

OWL COMPUTER SERVICES

5950 Keystone Drive
Bath, PA (215)-837-1917

Bits & Us - RadioShack® Dealer	*St. Onge Systems*	*Computers & Games*
Pottstown Ave., RT. 663	*Wescosville*	*Muhlenberg Shopping Plaza*
Pennsburg (215)-679-3389	*Call for Appt. (215)-481-9775*	*Reading (215)-929-0540*

DISK DRIVES

Bonus: Special Bundled Software with any Disk Drive Purchase!

Floppy Drive Systems

The Highest Quality for Years of Service

Drive 0 Systems (Half Height, Double Sided,

SOLD OUT!
WE NEED CONTROLLERS!

~~$189.~~

IF YOU HAVE 502 CONTROLLERS, CALL US!

Drive 1 Systems (Halt Height, Double Sided, Direct Drives) **$115.**

New 3.5", 720K Drives for OS-9 with case & Power Supply **$129. SALE!**

Drive 1 Systems have drive, case, power supply. (You may require optional cable and/or OOS chip to use)

Special for 0/1 Combos (0,1,2,3) $199.

(WITHOUT CONTROLLER)

HALF- HEIGHT DRIVE UPGRADES FOR RS HORIZONTAL CASES

Why only double the capacity of your system when you can triple in the same case? Kit includes: double-sided to fit your case, chip to run both sides of new drive, hardware, and detailed instructions. Easy! Takes only 5 minutes!

Model **Only $119.**

500 , 501, or 502

All drives are new and fully assembled. We ship only FULLY TESTED and CERTIFIED at these low prices. We use Fuji, YE Data, and other fine brands. No drives are used or surplus unless otherwise stated to you when you order. We appear to be the one of the few advertisers in Rainbow who can truly make this claim. We have 7 years experience in the CoCo disk drive market! We are able to provide support when you have a problem.

Drives 1 Year Warranty

OWL WARE Software Bundle

Disk Tutorial/Utilities/Games

DISK TUTOR Ver 1.1

Learn how to use your disk drive from this multi-lesson, machine language program. This tutor takes you through your lessons and corrects your mistakes for a quick, painless disk drive introduction. (This professionally written tutor is easily worth the bundle's total price.)

3 UTILITIES

A copy verify, copy, and DOS utility.

2 GAMES

We will select 2 games from our stock. These are sold for more than $20 each.

Do not mistake this software with cheap "Public Domain" software which others offer. All of this software is copyrighted and professional in quality. The tutor is unique with us and has helped thousands of new users learn their disk drive.

only $27.95 (or even better) only $6.95 with any Disk Drive Purchase!!

512K Upgrade

Again at a popular price. Fully assembled and tested before shipping. Easy to install. Uses fast 120 ns. chips.

SALE $79.

Now includes **memory test, Ram Disk Lighting, Printer Lighting, and Back-up Lighting.** All with an upgraded manual exclusive with OWL!

Our prices include a discount for cash but do not include shipping.

OWL WARE has a liberal warranty policy. During the warranty period, all defective items will be repaired or replaced at our option at no cost to the buyer except for shipping costs. Call our tech number for return. Return of non-defective or unauthorized returns are subject to a service charge.

OWL-WARE
P.O. BOX 116
Mertztown, PA 19539

Feature Program

Gopher SMASH

by THOMAS WONG

Have you ever tripped over a hole in your yard and discovered that gophers are inhabiting your property? If so, you probably won't be happy when the time comes to repair damages. While you're taking a break from your inevitable battle with the gophers, load *Gopher Smash* into your CoCo 3 and take out some of your frustrations.

After the program has initialized, you see nine boxes, each with a letter defining it. When a gopher pops up, press the key corresponding with the box containing the gopher, for as many times as the gopher is visible. The higher the gopher is above the ground, the more points you receive. Watch out for surprise bombs, though. The bombs are the same point value as the gophers except they subtract, rather than add, from your score.

The main goal is to gain the highest possible score in 60 seconds. When your time is up, you are asked if you want to try again or stop.

The game can be modified in several ways. For those who want customized keyboard layout, replace every third value of the DATA statement in Line 1 with the ASCII equivalents of the keys you want to use. Also, if the gophers pop up too fast, delete the high-speed poke in Line 1. These are just a few examples of what you can do to change the program. Good luck smashing gophers!

❑

CoCo 3

The Listing: SMASH

```
0 ' COPYRIGHT 1989  FALSOFT,INC
1 POKE65497,0:HSCREEN2:HCOLOR0,0
:HCOLOR4,0:H=0:K=1:FORA=1TO8:HBU
FFA,750:NEXTA:FORB=1TO9:READC(B)
,D(B),E(B):NEXTB:DATA32,40,81,12
8,40,87,224,40,69,32,88,65,128,8
8,83,224,88,68,32,136,90,128,136
,88,224,136,67
2 HCIRCLE(28,24),28.4,,3,.5,0:HC
IRCLE(14,24),8,4:HCIRCLE(14,24),
2,4:HCIRCLE(42,24),8,4:HCIRCLE(4
2,24),2,4:HDRAW"BM28,24F8L16E8BD
10R8D2G4L8H4U2R8D6":HPAINT(14,20
),7,4:HPAINT(42,20),7,4:HLINE(0,
24)-(0,40),PSET:HLINE-(56,40),PS
ET:HLINE-(56,24),PSET
3 HPAINT(28,28),2,4:HLINE(88,16)
-(88,20),PSET:HLINE(60,20)-(116,
40),PSET,B:HLINE(60,27)-(116,33)
,PSET,B:HPAINT(88,23),2,4:HPAINT
(88,30),3,4:HPAINT(88,36),5,4:HD
RAW"BM120,18D20R16U8R36U4L36U8L1
6":HPAINT(130,28),6,4:HLINE(120,
22)-(136,34),PSET,B
4 HGET(0,0)-(56,24),1:HGET(0,8)-
(56,32),2:HGET(0,16)-(56,40),3:H
GET(180,16)-(236,40),4:HGET(60,0
)-(116,24),5:HGET(60,8)-(116,32)
,6:HGET(60,16)-(116,40),7:HGET(1
20,16)-(176,40),8:HCLS:HCOLOR2:H
LINE(4,4)-(316,28),PSET,8F:HLINE
(4,32)-(316,182),PSET,B
5 HCOLOR3:FORF=16TO208STEP96:FOR
G=65TO161STEP48:HLINE(F,G)-(F+88
,G+16),PSET,BF:NEXTG,F:HCOLOR4:H
PRINT(7,9),"Q":HPRINT(19,9),"W":
HPRINT(31,9),"E":HPRINT(7,15),"A
":HPRINT(19,15),"S":HPRINT(31,15
),"D":HPRINT(7,21),"Z":HPRINT(19
,21),"X":HPRINT(31,21),"C
6 I=0:J=61:HPUT(C(K),D(K))-(C(K)
+56,D(K)+24),4
7 J=J-1:HPRINT(1,1),"SCORE:":HPR
INT(19,1),"HI SCORE:":HPRINT(16,
2),"TIME:":HCOLOR2:HLINE(56,8)-(
136,15),PSET,BF:HLINE(232,8)-(30
4,15),PSET,BF:HLINE(168,16)-(190
,23),PSET,8F:HCOLOR4:HPRINT(7,I)
,I:HPRINT(28,1),H:HPRINT(21,2),J
8 IF I>H THENH=I:J=J+1:GOTO7
9 IF J<1 THEN HPRINT(15,5),"GAME
 OVER":HPRINT(4,23),"Do you want
 to play again (Y/N)?":R$=INKEY$
:IFR$="Y"THENHCOLOR0:HLINE(120,3
8)-(192,47),PSET,BF:HLINE(32,184
)-(288,192),PSET,BF:HCOLOR4:GOTO
6ELSEIFR$="N"THENPOKE&HFFD8,0:W1
DTH32:ENDELSE9
10 K=RND(9):L=RND(2):IF L=1THENM
=0ELSEM=4
11 N=M+1:O=N:P=1
12 HPUT(C(K),D(K))-(C(K)+56,D(K)
+24),O:O$=INKEY$:IF O$<>"" THEN
GOSUB17
13 O=O+P
14 IF O>N+2 THEN O=N+2:P=-1
15 IF O<N THEN HPUT(C(K),D(K))-(
C(K)+56,D(K)+24),4:GOTO7
16 GOTO12
17 FORS=1TO9
18 IF ASC(O$)=E(S)THEN19ELSE20
19 HPUT(C(S),D(S))-(C(S)+56,D(S)
+24),8:HPUT(C(S),D(S))-(C(S)+56,
D(S)+24),4:SOUNDE(S),1:IF S=K TH
EN GOSUB21
20 NEXTS:RETURN
21 IFI=1THENI=I+0:HPRINT(1,2),"O
UCH!":GOSUB24:HLINE(8,16)-(48,24
),PSET,BF
22 IFL=2THENI=1-(0-4):HPRINT(34,
2),"BOOM!":GOSUB24:HLINE(272,16)
-(312,24),PSET,BF
23 HCOLOR4:RETURN
24 HCOLOR2:FORT=1TO50:NEXTT:RETU
RN
```

Feature Program

Prompt Improvement

by Ric Pucella

When I am working in BASIC, I find it convenient to know which drive is currently selected and how much free space I have in memory. The short program shown here modifies the BASIC interpreter to display this information every time the OK prompt is displayed.

Enter the program as shown and save it to tape or disk. When you run it, a short machine-language routine that handles the modification is installed in memory. After this, you'll see the drive number and free memory displayed. (Users with tape-based systems will see a drive number, though it won't really mean anything.)

Once the program has been run, you can save the machine-language portion to disk by entering

```
SAVEM"PROMPT",&H7F00,&H7F47,&H7F
00
```

Afterward, you can enter the following two commands to change your prompt:

```
CLEAR 200,&H7EFF
LOADM"PROMPT":EXEC
```

(Tape users should change SAVEM and LOADM above to CSAVEM and CLOADM.)

While the modification is in place, don't press Reset. Doing so removes the modification and you'll have to run the program again. Also, while intended for the CoCo 3, the program works on the Coco 1 and 2 as long as the computer is put in the all-RAM mode first.

❑

CoCo 3

The Listing: PROMPT

```
1 'PROMPT IMPROVEMENT
2 'BY RIC PUCELLA
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
60 CLEAR200,&H7F00-1
70 L=&H7F00
80 GOSUB100
90 CLS:PRINT:PRINT:PRINT"PROMPT
CHANGED...":PRINT:PRINT:EXEC&H7F
00:NEW
100 READA$:FOR1=1TOLEN(A$)STEP2:
B$=MID$(A$,1,2):IFB$="**" THEN R
ETURN ELSE POKEL,VAL("&H"+B$):L=
L+1:NEXT:GOTO100 'ML LOADER
110 DATA8E7F07BFAC7A39BDB9588E7F
2EBDB99C86095A8B30BDA2828E7F3ABD
B99C1F40931FBDB0CCBDB95B8EABEDBD
B99C392020444546204452202320002O
20465245452040454D200020**
```

Feature Program

HSCREENs: Cut 'em Down to Size

by Joel Mathew Hegberg

Let's face it, HSCREEN picture files take up way too much storage space. Some would argue that it's a small price to pay for the advanced graphics resolution and color capabilities the Color Computer 3 provides.

But each image eats up 16 granules of disk space, you can store only four pictures on a standard 35-track single-sided disk! When I think about this, I can't help but remember the days of PMODE screens, which occupied a mere three granules each. Wouldn't it be nice to have some way to shrink HSCREEN picture files in an effort to achieve this same level of disk-space conservation? Now there is a way — *CompSaver* and *CompLoader*.

CompSaver is a graphics utility that compresses and saves HSCREEN images to disk. Its brother program, *CompLoader*, handles loading these compressed images, decompressing them at load time. In my experience, 90 percent of HSCREEN images are not too detailed. These pictures can usually be compressed down to six granules or less. The other 10 percent contain more detail and take up anywhere from 10 to 18 granules.

To get started, enter the programs shown in listings 1 and 2, and save them to disk as CMPSAVE.BAS and CMPLOAD.BAS, respectively. It is important that you save them before you run them, especially if you have made any modifications or corrections. When executed, these programs poke machine-language routines into memory. Both programs check the data statements for errors as they perform this task.

The picture to be compressed must first be loaded into memory. Any image created by a BASIC program should not present a problem, and other images can be loaded using one of the many viewers we've seen over the years. There are some things to remember, though. Since *CompSaver* saves the palettes along with the image, make sure you don't alter the palette settings by entering RGB or CMP, or by pressing Reset.

Once the image is in memory, run CMPSAVE (Listing 1). When you are prompted to enter the name of the picture, enter a standard filename (up to eight characters) with no extension. You are then asked to enter the HSCREEN in which the image is stored in memory. Enter 1, 2, 3 or 4, accordingly. (Most CoCo 3 images are stored in HSCREEN 2.) After this, your picture is saved to disk by a machine-language routine. (I used machine language because of its tremendous speed advantages over BASIC for this type of task.)

Loading a compressed image is easy, too. Run CMPLOAD (Listing 2) and enter the name of the file. The machine-language loading routine takes over and, after the picture is loaded, it is displayed on the screen.

The machine-language routines in CMPSAVE and CMPLOAD are completely relocatable. If you have experience with assembly, feel free to move these routines around if they cause conflicts with one of your own machine-language creations.

Now you can enjoy all those HSCREEN pictures without breaking the bank buying disks.

Joel Mathew Hegberg has been programming for nine years and enjoys writing software for the Color Computer and the MM/1. Some of his creations are commercially available through Sub-Etha Software. Joel may be contacted at 936 N. 12th St., DeKalb, IL 60115-2516, (815) 748-6638. Please include an SASE when requesting a reply.

Reviewer Information

In order to continue to bring Tandy Color Computer users all the best information about new hardware and software products each month, we are constantly looking for new people to join our independent review staff. Therefore, we invite you to join THE RAINBOW's elite fleet of reviewers.

You read THE RAINBOW because you love your Color Computer, so if you want a creative outlet and a chance to examine quality hardware and software, with your observations published nationwide, we want to hear from you.

Send us a cover letter with your name, address, occupation, list of equipment, areas of general interests, and a sample review of a CoCo product you are currently using. We look forward to your response. After all, we already see you have the best taste in computers.

Reply to: Reviews Editor, The Rainbow, The Falsoft Building, P.O. Box 385, Prospect, KY 40059

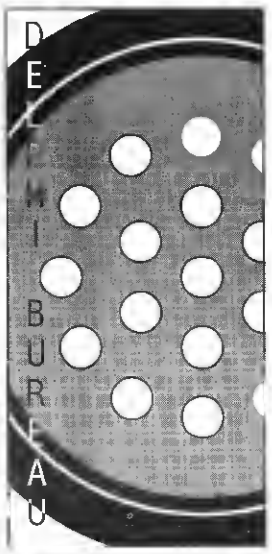
CoCo 3

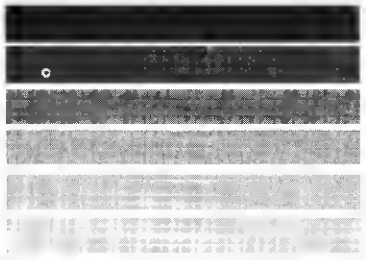
Listing 1: CMPSAVE

```
1 'COMPRESSED HSCREEN SAVER
2 'BY JOEL HEGBERG
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 '
20 ' COMPRESSED HSCREEN SAVER
30 ' BY JOEL MATHEW HEGBERG
40 ' 936 NORTH TWELFTH STREET
50 ' DE KALB, IL  60115
60 '
70 '
80 CLEAR1000,31000 'DEFAULT ADDR
ESS 31000
90 RESTORE:LC=31000 'DEFAULT LOC
ATION 31000
100 LL=0:TL=0:LN=1000:LK=LC
110 READ A$:IFA$="**"THEN 160
120 IFLEN(A$)<=2 THEN V=VAL("&H"
+A$):POKE LC,V:LL=LL+V:LC=LC+1:G
OTO 110
130 V=VAL(A$):IF V=LL THEN TL=TL
+LL:LL=0:LN=LN+10:GDTOII0
140 PRINT"ERRDR WITH DATA IN":PR
INT"LINE #";LN
150 STDP
160 READ A$:V=VAL(A$):TL=TL+LL
170 IF TL=V THEN 200
180 PRINT"ERROR IN DATA STATEMEN
TS."
190 STDP
200 CLS:PRINT"ENTER NAME OF PICT
URE."
210 LINE INPUT">";NM$:NM$=NM$+".
CPS"
220 PRINT"WHICH HSCREEN? (1-4)"
230 INPUT HS:HS=INT(HS):IF HS<1
OR HS>4 THEN SDUND1,5:GDTO200
240 OPEN"D",#1,NM$:PDKE LK,HS:PD
KE LK+1,0
250 EXEC LK+2
260 CLOSE #1:SDUND200,3
270 PRINT NM$;" IS SAVED."
280 END
1000 DATA 2,0,6F,8D,0,DF,8E,40,0
,AF,8D,0,D9,8E,FF,B0,C6,10,86,21
37
1010 DATA 1,97,6F,A6,8C,E7,AD,9F
,A0,2,A6,8C,E1,AD,9F,A0,2,A6,80,
2613
1020 DATA 34,14,AD,9F,A0,2,35,14
,5A,C1,0,10,22,FF,EF,86,4A,AD,18
47
1030 DATA 9F,A0,2,86,4D,AD,9F,A0
,2,86,48,AD,9F,A0,2,A6,8D,0,9A,2
187
1040 DATA 1A,50,8B,70,B7,FF,A2,8
6,1,A7,8D,0,91,AE,8D,0,8A,A6,80,
2292
1050 DATA A7,8D,0,86,8C,5F,FF,10
,22,0,23,A6,8D,0,7C,8I,FD,10,22,
1880
1060 DATA 0,19,A6,B4,A1,8D,0,6F,
10,26,0,F,A6,B0,A6,8D,0,66,88,16
47
1070 DATA 1,A7,8D,0,60,I6,FF,D6,
AF,8D,0,56,C6,1,D7,6F,A6,8D,0,21
30
1080 DATA 5I,AD,9F,A0,2,A6,8D,0,
48,AD,9F,A0,2,AE,8D,0,3E,BC,60,2
061
1090 DATA 0,10,25,FF,98,A6,80,0,
32,BB,1,A7,BD,0,2C,81,3,10,22,14
91
1100 DATA 0,1E,8E,40,0,AF,8D,0,2
0,A6,8D,FF,36,81,1,10,27,0,3,138
8
1110 DATA 16,FF,74,A6,8D,0,E,81,
1,10,23,FF,6A,86,7A,87,FF,A2,211
2
1120 DATA 1C,AF,39,FF,FF,FF,FF,0
,**,25065
```

Listing 2: CMPLOAD

```
1 'COMPRESSED HSCREEN LDADER
2 'BY JOEL HEGBERG
3 'CDPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBDW MAGAZINE
10 '
20 ' COMPRESSED HSCREEN LDADER
30 ' BY JDEL MATHEW HEGBERG
40 ' 936 NDRTH TWELFTH STREET
50 ' DE KALB, IL  60115
60 '
70 '
80 CLEAR1000,31000 'DEFAULT ADDR
ESS 31000
90 RESTORE:LC=31000 'DEFAULT LDC
ATION 31000
100 LL=0:TL=0:LN=1000:LK=LC
110 READ A$:IFA$="**"THEN 160
120 IFLEN(A$)<=2 THEN V=VAL("&H"
+A$):POKE LC,V:LL=LL+V:LC=LC+I:G
OTO 110
130 V=VAL(A$):IF V=LL THEN TL=TL
+LL:LL=0:LN=LN+10:GOTO110
140 PRINT"ERRDR WITH DATA IN":PR
INT"LINE #";LN
150 STOP
160 READ A$:V=VAL(A$):TL=TL+LL
170 IF TL=V THEN 200
180 PRINT"ERRDR IN DATA STATEMEN
TS."
190 SIOP
200 CLS:PRINT"ENTER NAME OF PICT
URE."
210 LINE INPUT">":NM$:NM$=NM$+".
CPS"
220 OPEN"I",#1,NM$:EXEC LK+2
230 HSCREEN PEEK(LK)
240 EXEC LK+23
250 CLOSE #1:SOUND 200,3
260 GOTO260
1000 DATA 2,0,6F,8D,0,AC,8E,40,0
,AF,8D,0,A6,1A,50,C6,1,D7,6F,I7,
1/68
1010 DATA 0,8D,39,17,0,6A,B0,A1,
76,81,0,10,27,0,59,A7,8D,0,8E,15
I8
I020 DATA A6,8D,0,87,8B,70,B7,FF
,A2,BD,A1,76,E6,8D,0,7E,AE,8D,25
73
1030 DATA 0,78,A7,80,5A,C1,0,10,
22,FF,F7,AF,8D,0,6B,8C,60,0,10,1
925
I040 DATA 25,FF,C8,A6,8D,0,5F,8B
,1,A7,8D,0,59,81,3,10,22,0,1D,16
45
1050 DATA 8E,40,0,AF,8D,0,4D,A6,
BC,97,81,1,10,27,0,3,I6,FF,A8,16
89
1060 DATA A6,8D,0,3C,81,1,I0,23,
FF,9E,86,7A,B7,FF,A2,IC,AF,39,20
77
I070 DATA 8E,FF,B0,C6,10,34,14,B
D,A1,76,35,14,A7,80,5A,C1,0,10,1
994
1080 DATA 22,FF,F0,BD,A1,76,8D,A
1,76,8D,A1,76,39,8D,A1,76,A7,8D,
2766
1090 DATA FF,56,BD,A1,76,A7,8D,F
F,50,39,**,19464
```





EDDIE KUNS

Editing Forum Messages

Do you know that you can edit any messages you post in Forum? This comes in really handy whether you simply want to correct a few typing errors or need to change something that is much more important. Suppose a Forum thread has strayed from its original subject — the original message asked a question about using RS-232 ports under OS-9 but you get sidetracked discussing the merits of a specific terminal program. To keep other users from getting confused, it is a good idea to edit the message's Subject to reflect this change. This is also courteous, especially to those who are searching through the Forum looking for messages discussing specific topics.

Once you have posted a message, you can edit any part of it. While you can edit any message you have posted, you cannot edit another user's message. To get started, enter the following at the Forum prompt:

```
EDIT message-number
```

where *message-number* is the number of the message you want to edit. The Edit menu (shown in Figure 1) is displayed. As with many other Forum commands, you can enter EDIT by itself to edit the current message. Enter EDIT ? to see the many other options it supports.

To change the subject of the message you are editing, enter SUBJECT (or an appropriate abbreviation) at the EDIT> prompt. Delphi prompts you to enter the new subject. Do so, or press ENTER by itself to retain the current subject. To also change the topic of the message, enter TOPIC at the EDIT> prompt and follow a similar procedure. If you forget what the current messages's subject and topic are, enter SHOW to display the message header and the first line or so of the message. You can also delete a Forum message from the Edit menu (by entering DELETE or DEL), although it's much easier to use the DELETE command directly from the FORUM> prompt.

```
FORUM EDIT Menu:

TEXT of Current Message
SUBJECT of Current Message
TOPIC of Current Message
DELETE Current Message
SHOW Message Header
HELP
EXIT
```

Figure 1: Forum Edit Menu

If you want to edit the body text of the message, enter TEXT at the EDIT> prompt. This drops you into your selected editor (EDT or Oldie). Within the editor, you can change all the text of the Forum message, much as you might use your editor to edit a file in Workspace. If you chosen editor is EDT, you'll need to enter EXIT to return from the editor to the EDIT> prompt. Users of Oldie should enter /E to exit.

When you have finished editing the Forum message, press CTRL-Z at the EDIT> prompt (or enter EXIT) to return to the FORUM> prompt.

Eddie Kuns is pursuing a doctorate in physics at Rutgers University. He lives in Aurora, Illinois, and works as a programmer and researcher at Fermilab. Eddie is the database manager of the OS-9 SIG and can be reached online as EDDIEKUNS.

The Delphi Voting Booth

Two areas on Delphi that appear to be underused are Poll in the CoCo SIG and Voting Booth in OS9 Online. Although their names differ, these areas have exactly the same function: they let you "speak your piece." Once you enter the Poll area, you'll see the POLL> prompt, where you have the options shown in Figure 2.

To see a list of the polls on which you can vote, simply enter LIST. All active polls will be displayed. To see the results of a specific poll, enter RESULTS *poll name*. (If you enter RESULTS by itself, you are prompted for the poll name.) You will see something like the text shown in Figure 3. After the votes for that poll are displayed, Delphi lists the comments voters added. Finally, you are prompted for whether or not you want to vote on that poll, even if you've already voted on it. If you answer Yes but have already voted on the Poll, Delphi asks if you want to change your previous vote. (No, Delphi doesn't support the Chicago-style "vote early, vote often" approach.)

Another way to vote on a specific poll is to use the VOTE command. As with the RESULTS command, you can either provide the poll name on the command line or let Delphi prompt you. After you enter your vote, Delphi allows you to enter a comment on the poll. Your comment is limited to about four 80-character lines.

If you have already voted on a poll and don't want to change your vote, but you do want to change your comment, use the EDIT command. If you created the poll, you also use EDIT to modify it.

The BROWSE command allows you to travel through each poll in sequence, giving you a chance to see how others voted. When you enter BROWSE, you start at the first active poll and end with the last. Of course if you want to leave Browse early, press CTRL-Z.

You can create your own poll by using the CREATE command. When you enter CREATE, you are prompted for a poll name. Enter something that people will understand when they use the LIST command. Next, you are prompted for the poll type. There are three kinds of polls: 1) Yes/No; 2) strongly agree through strongly disagree (the person answering the poll has five choices); and 3) multiple choice, in which you enter specific choices, as Jim Reed did in the example.

If you choose to create a multiple-choice poll, Delphi allows you to enter up to 12 categories, each of which may be up to 20 characters in length. Enter your choices and press CTRL-Z. You can even opt to allow voters to add new categories when they vote. But remember that there can be only 12 categories altogether.

Finally, you are asked to enter your argument. The text you enter appears before the actual poll. In the sample poll from the CoCo SIG, Jim Reed started entering the text at "I was sitting here . . ."

Once you have created a poll, by all means vote on it. After all, you must have a reason for having created the poll in the first place. Then feel free to enter the Forums of both SIGs and advertise your poll.

```
BROWSE through poll results      HELP
CREATE a new poll                LIST poll names
EDIT your poll comment           RESULTS with comments
EXIT                             VOTE on a poll
```

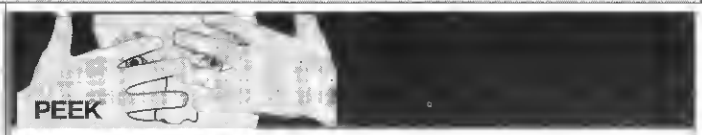
Figure 2: Poll Menu

```
SCREEN VIEWING DISTANCE, created by JIMREED.
Creation date: MAY 16,1989

I was sitting here staring at the screen a
bit closer than normal for me and
began wondering how far from the screen
most people sit. I was thinking about
something and realized I was only 12 inches
from the screen. Usually, I sit
about 16 inches away - not that far! So,
when you are using the computer, how
far away are your eyes from the screen
surface?

CHOICE                  VOTES   PERCENT
------                  -----   -------
Less than 12 inches         2        2%
12 to 14 inches             1        1%
14 to 16 inches             1        1%
16 to 18 inches             9       11%
18 to 20 inches            13       16%
20 to 22 inches            15       18%
24 to 30 inches            32       40%
30 to 36 inches             5        6%
over three feet!            1        1%
-----
TOTAL VOTE:                79
```

Figure 3: Sample Poll



With the CoCo 1 and 2, it is especially important to make sure the printer is online before sending data to it. Include the following line in your BASIC programs to determine whether or not the printer is ready:

```
A=(PEEK(65314) AND 1)
```

If the value returned in Variable A is 1, the printer is *not* ready. (You can use any numeric variable in place of A.)

Conference Schedule

Several regularly scheduled conferences take place in the CoCo and OS9 Online SIGs. On the first Monday of each month, Steve Bjork discusses game programming, and on the second Monday of each month, Tim Kientzle and I have a conference about using Delphi.

In addition to the two monthly conferences, there are four weekly conferences:

Thursday — OS-9 Help Line
led by Chris Deierlein
Thursday — RiBBS Help/Talk Conference
led by Charles West
Friday — The Art and Science of UUCP
led by Rick Adams and Trix
Saturday — AcBBS
led by Chris Serino and the authors of AcBBS

All regularly scheduled conferences take place at 10 p.m. Eastern time. There are also many spontaneous conferences. Remember that anyone whose name is surrounded by parenthesis when you do a /WHO command is in the conference area.

Uploads at a Glance

In the OS9 Online General Information database, **Michael Dalene** (MDALENE) posted a demo for the Star-Gemini NX-1020 Rainbow printer, showing off the printer's features as well as demonstrating how to integrate escape codes into a text file using the VI editor. If you want to look through the databases to see what's there but don't want to spend several hours online searching file by file for something interesting, download **Greg Law**'s (GREGL) contribution: 15 files listing all the groups (with descriptions) in each database topic. **Paul Wright** (PWRtGHT) posted transcripts for an AcBBS conference as well as a C conference.

In the Applications database topic, **Paul M. Fitch, Jr.** (EMTWO) released a new `error` command that works with the `help` command **Tim Kientzle** (TIMKIENTZLE) posted a couple of years ago. The new `error` command prints verbose descriptions of OS-9 error numbers. In the Telecom (6809) database, **Ken Flanagan** (KENFLANAGAN) released the latest version of *Scribe*, a program that allows you to read mail messages offline when you receive `.QWK` packets.

If you've been having trouble using the *PCDOS* version of `CC3D1sk` together with Bruce Isted's serial-mouse patch for the CoCo, you'll be interested in Jim Martin's file describing how to fix the interrupt conflict between the two devices. In the Programmers Den topic, **Robert Kemper** (BOBKEMPER) released an archive of information to help BASIC09 programmers. **Don Berrie** released *CENV* — a point and click environment for the C compiler on the CoCo.

In the OSK Applications database, **Eric Crichlow** (HYPERTE) released *Image Master*, an icon and sprite editor designed for the MM/1 under *KWwindows*. If you have never used `make` to maintain a program but want to, take a look at the examples **Glen Hathaway** (COMPER) posted in the Tutorials database.

In the CoCo SIG CoCo 3 Graphics database, **Chet Simpson** (HYPERTECH) released a new version of *Image Master*. This version is customized for the special features of the 6309 and uses block moves to really speed up things. In the Utilities & Applications database, **M. David Johnson** (MDJOHNSON) released a number of utilities designed to work with CF83 — a version of Standard Forth '83. **Richard McNabb** (RICKMAC) released a new version of `DIRU3`, which allows copying files between disks and many other disk-maintenance functions.

In the Games database, **Johnny Williams** (DRILLMASTER) uploaded a slot-machine program for the CoCo 3.

DATABASE REPORT

OS-9 SIG

General Information
- STAR NX-1020 SHOWS OFF — MDALENE — Michele Dalene
- SCSI ADAPTOR ANOUNCEMENT — FltOGG — Frank Hogg
- DATABASE LISTINGS — GREGL — Greg Law
- OS-9 COMMUNITY NETWORK INFO, AUG — AtRWOLF2 — Greg Morgan
- CONFRFNCE TRANSCRIPTS — PWRtGHT — Paul Wright

Applications (6809)
- ERROR/HELP:REPLACEMENT ERROR CMD — EMTWO — Paul M. Fitch, Jr.
- STREAM: HARD DRIVE BACKUP UTIL — JENG — John Eng
- GETIME: CLOCK SETTING UTIL — BLAINET — Blaine Tempest
- REPACK FIX FOR KRN_ PATCH — COCOXT — Christopher Burke
- DBL 2: DOUBLE SIDED PRINTER — WOAY — Jim Martin

Telecom (6809)
- SCRIBE 4.0 OFFLINE READER — KENFLANAGAN — Ken Flanagan

System Modules (6809)
- SMOUSE INTERUPT CLASH FIX — WOAY — Jim Martin

Games & Graphics
- STRIP POKER FOR OS9 — DEANHOLDER — Dean Holder
- PIXSHOW - MM/1 .PIX VIEWER — BRUCEGERST — Bruce Gerst
- HANGMAN V2.4 — MOHRT — Tim Mohr

Music & Sound
- BLADE RUNNER: STEREO SOUND — JOHNBAER — John Baer

Programmers Den
- BASIC09 HELP FILES — BOBKEMPER — Robert Kemper
- CENV:C COMPILER USER INTERFACE — DABERRIE — Don Berrie

OSK Applications
- PSF: TEXT TO POSTSCRIPT CONV. — MARKGRIFFITH — Mark Griffith
- DVI2EP2: DVI TO EPSON CONVERTER — JOHNREED — John Wainwright
- IMAGE MASTER — HYPERTE — Eric Crichlow
- GNU INDENT EXECUTABLE — NIMITZ — David Graham
- TSTART: COLORIZE TASCOM — COMPER — Glen Hathaway
- DIRECTORY COPIER — COMPER — Glen Hathaway

Tutorials & Education
- ELM INSTALLER SH-SCRIPT — THEFERRET — Philip Brown
- SAMPLE MAKE FILES — COMPER — Glen Hathaway

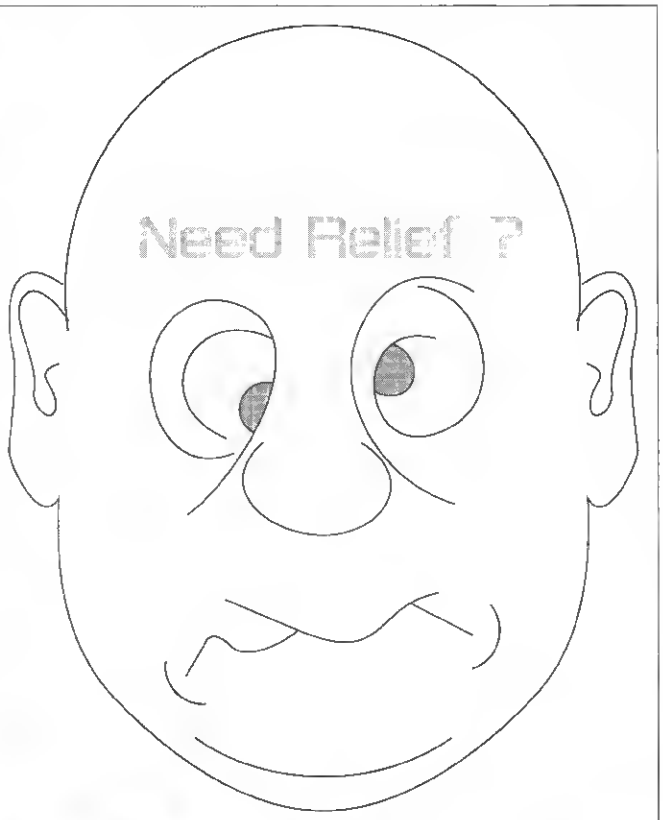
Standards
- IFF FILE SPECIFICATIONS — MDALENE — Michele Dalene

CoCo SIG

CoCo 3 Graphics
- IMAGE MASTER V4.0 — HYPERTECH — Chet Simpson

Utilities & Applications
- CF83 BLOCK/FILE CONVERSIONS — MDJOHNSON — M. David Johnson
- CF83 PMODE GRAPHICS TEXT CHARACT — MDJOHNSON — M. David Johnson
- CF83 BENCHMARK — MDJOHNSON — M. David Johnson
- CF83 TEXT SCREEN CONTROL — MDJOHNSON — M. David Johnson
- DIRECTORY TO ASCII FILE. — MDJOHNSON — M. David Johnson
- TWO DISK DIR/FILE UTIL UPDATE — RICKMAC — Richard McNabb

Games
- SLOTCOCO.BAS — DRILLMASTER — Johnny Williams

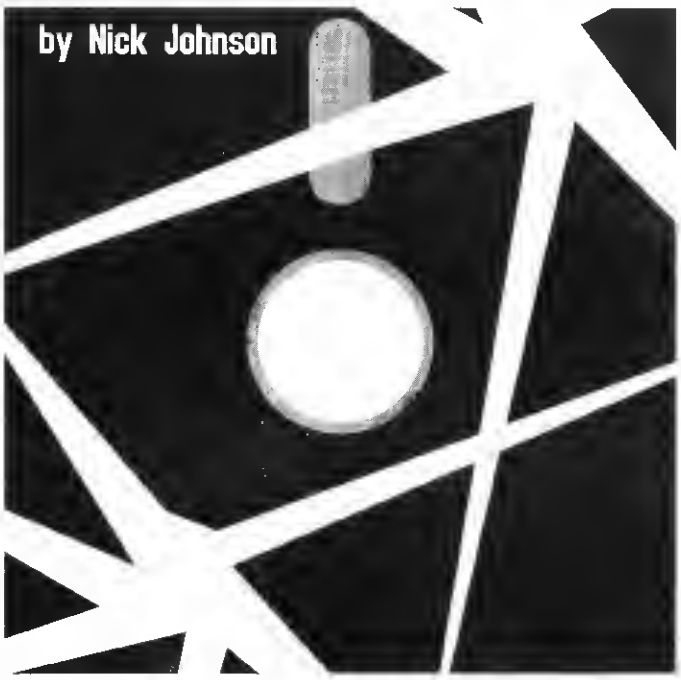


THE RAINBOW TO THE RESCUE

SEE PAGE 14

Feature Program

NO MORE DISK FRAGMENTS!



Disk Extended Color BASIC is antiquated in that it includes many compensations for the unreliable hardware of its time. These include long timing delays (to allow the motor to come up to speed) and intentional fragmentation of disk data (to prevent undue wear on the disks).

When was the last time you noticed "wear" on one of your disks? Chances are, you haven't. Yet Disk BASIC insists on scattering file data across your disks in an attempt to spread the "wear" as thinly as possible. It does this by storing files from the directory track (Track 17) outward (toward the outside *and* inside edges of the media), alternating on a granule-by-granule basis.

The unfortunate result of all this is an increase in access time, especially for larger files. This decrease is even more noticeable if the files are frequently used.

To overcome this decrease in performance, I wrote *CoCo Disk Defragmentor*. This program takes the bits and pieces of the files on a disk and puts them *together*, storing the result on a new disk. In the process, it rewrites the granule table.

CoCo Disk Defragmentor requires two disk drives and works with any CoCo having at least 32K of memory. If you use a CoCo 3, the computer is set to 32-column screen (if it is not already there) and Super Extended BASIC is disabled (the "CoCo-2 mode"). To regain access to the CoCo 3 functions (perhaps to run a listing), enter

```
CLEAR 200,32768:POKE 65503,0
```

To use *CoCo Disk Defragmentor*, first enter the program exactly as it appears in the listing, and save it to disk. When you enter RUN, a menu appears showing you the program's three options: Begin Defragmentation, Display Granule Table and Quit.

To defragment a disk, press B. Make sure the disk you want to defragment is write protected (to prevent data loss should a crash occur). Now insert the disk into Drive 0, put a blank formatted disk in Drive 1, then press ENTER. As the program goes to work, the screen displays the current filename, operation, track and sector, granule number and drive number. This allows you to monitor the operation. Depending upon the number of files on the original disk, defragmentation may take from one to 10 minutes. Finally, never reuse or get rid of the original disk. You may need it later for backup purposes.

The Display Granule Table option shows you the granule table for the disk in Drive 0. Granules are numbered from 0 to 67, and each entry in the table points to the next granule in the file. Granules that begin with a C (as in hexadecimal C4) indicate how many sectors are used in the last granule of the file. You can use this option to examine the effects of defragmentation.

Programs that expect to find certain information on specific sectors of the disk won't function properly when they are defragmented; this will be encountered more often with machine-language programs than with BASIC. Defragmentation is basically an organized COPY — it does not back up the entire disk — and *CoCo Disk Defragmentor* does not know these programs have special requirements.

I hope you enjoy using this handy little utility. If you have any comments, suggestions or questions, feel free to write.

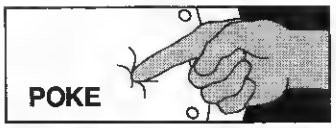
Nick Johnson is 17 years old and is a senior at Crestview High School, where he participates in the gifted student program. He started programming on a 32K CoCo I in 1982 and, after purchasing a CoCo 3, advanced quickly; in his own words, Nick "now programs almost constantly." He may be contacted at 5830 Reinke Dr., Crestview, FL 32536-8913. Please include and SASE when requesting a reply.

CoCo 3

The Listing: DISKERAG

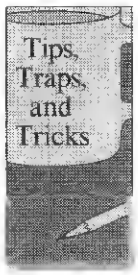
```
1 'DISK DEFRAGMENTOR
2 'BY NICK JOHNSON
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
10 VERIFYON
20 IF PEEK(&HFFFF)=27 THEN WD=PE
EK(&HE7):EXEC &HF652:POKE 65502,
0
30 PCLEAR 1
40 OV=PEEK(&H95A)
50 CLEAR 20000:DIM F1$(68):DIM A
$(10):DIM B$(10):DIM G(72)
60 CLS
70 PRINT"  -=> COCO DISK DEFRAGM
ENT <=-"
80 PRINT"  -=>   BY NICK JOHNSON
   <=-"
90 PRINT:PRINT
100 PRINTTAB(6)"BEGIN DEFRAGMENT
ATION"
110 PRINT
120 PRINTTAB(6)"DISPLAY GRANULE
TABLE"
130 PRINT
140 PRINTTAB(6)"QUIT"
150 PRINT@100,STRING$(25,207):PR
INT@292,STRING$(25,207)
160 FOR X=4 TO 8:PRINT@(X*32)+4,
CHR$(207);:PRINT@(X*32)+28,CHR$(
207)::NEXT X
170 PRINT@384
180 A$=INKEY$:IF A$="" THEN 180
190 IF INSTR("BDQ",A$)=0 THEN 18
0
200 IF A$="B" THEN 220 ELSE IF A
$="D" THEN 1200 ELSE IF A$="Q" T
HEN 210
210 GOSUB 1810:CLEAR 200,32768:I
F PEEK(&HFFFF)=27 THEN POKE 6550
3,0:POKE &HE7,WD:END
220 CLS
230 A$="DEFRAGMENTATION"
240 GOSUB 1350
250 PRINT"INSERT A BLANK, FORMAT
TED DISK  IN DRIVE 1."
260 PRINT:PRINT"INSERT THE FRAGM
ENTED DISK IN   DRIVE 0."
270 PRINT:PRINT"PRESS [enter] WH
EN READY."
280 IF INKEY$<>CHR$(13) THEN 280
290 CLS
300 A$="DEFRAGMENTATION IN PROGR
ESS"
310 GOSUB 1350
320 PRINT:FAT$=""
330 PRINTTAB(5)"PRESS [ENTER] TO
 ABORT."
340 PRINT@160,"FILENAME"
350 PRINT@192,"CURRENT OP:"
360 PRINT@224,"TRACK, SECTOR"
370 PRINT@256,"GRANULE"
380 PRINT@288,"DRIVE"
390 PRINT
400 F$="":OP$="READING DIRECTORY
":T=-1:S=-1:G=-1:D=0
410 GOSUB 1480
420 'READ THE DIRECTORY.
430 GOSUB 1810
440 FOR I=3 TO 11
450 DSKI$0,17,1,A$(I-1),B$(I-1)
460 S=1:T=17:GOSUB 1480
470 NEXT I
480 GOSUB 1790
490 OP$="PROCESSING DIR":T=-1:D=
-1:GOSUB 1480
500 P=2
510 N=1:O=1
520 IF (N-1)*32>=128 THEN 570 EL
SE Q$=MID$(A$(P),(N-1)*32+1,32)
530 IF ASC(Q$)=0 THEN N=N+1:GOTO
 520
540 IE ASC(A$(P))=255 OR ASC(Q$)
=255 THEN 650
550 F1$(O)=Q$
560 IF (N-1)*32>=128 THEN 570 EL
SE N=N+1:O=O+1:GOTO 520
570 N=1
580 IF (N-1)*32>=128 THEN 630 FI
SEQ$=MID$(B$(P),(N-1)*32+1,32)
590 IF ASC(Q$)=0 THEN N=N+1:GOTO
 580
600 IF ASC(B$(P))=255 OR ASC(Q$)
=255 THEN 650
610 F1$(O)=Q$
620 IF (N-1)*32>=128 THEN  630 E
LSE N=N+1:O=O+1:GOTO 580
630 P=P+1:IF P>11 THEN 650
640 N=1:GOTO 520
650 '
660 O=O-1 'O IS # OF FILES
670 DP$="READING FAT"
680 T=17:S=2:D=0
690 GOSUB 1480
700 GOSUB1810:DSKI$0,17,2,A$(1),
B$(1)
710 GOSUB1790:GR=0 'LAST AVAILAB
LE GRN ON NEW DISK
720 FOR K=1 TO O 'OUTER LOOP
730 IF INKEY$=CHR$(13) THEN 50
740 F$=LEFT$(F1$(K),11)
750 OP$="ANALYZING FILE":T=-1:D=
-1:G=-1:GOSUB 1480
760 FG=ASC(MID$(F1$(K),14,1)):EG
=EG '1ST GRAN
770 LB=ASC(MID$(F1$(K),15,1))*25
6+ASC(MID$(F1$(K),16,1))
780 G(K)=GR
790 OP$="GRABBING GRANULE"
800 D=0:G=EG:GOSUB 1550
810 OP$="WRITING GRANULE."
820 D=1:G=GR:GOSUB 1680
830 DP$="CHECKING NEXT GR":D=-1:
T=-1:S=-1:G=-1:GDSUB 1480
840 NG=ASC(MID$(A$(1),EG+1,1))
850 IF NG>=192 THEN 930
860 EG=NG:GR=GR+1
870 DP$="UPDATING FAT":D=1:T=17:
S=2:G=-1:GOSUB 1480
880 FAT$=FAT$+CHR$(GR)
890 GOSUB 1810
900 DSKO$1,17,2,FAT$,""
910 GOSUB 1790
920 GOTO 790
930 OP$="UPDATING FAT":D=1:T=17:
S=2:G=-1:GOSUB 1480
940 FAT$=FAT$+CHR$(NG):GR=GR+1
950 GOSUB 1810
960 DSKO$1,17,2,EAT$,""
970 GOSUB 1790
980 NEXT K
990 DP$="UPDATING DIR":D=1:G=-1
1000 '
1010 F1=1:FOR MS=2 TO 10
1020 FOR I=14 TO 128 STEP 32
1030 MID$(A$(MS),I,1)=CHR$(G(FI)
)
1040 F1=F1+1
1050 NEXT I
1060 FOR I=14 TO 128 STEP 32
1070 MID$(B$(MS),I,1)=CHR$(G(FI)
)
1080 F1=F1+1
1090 NEXT I:NEXT MS
1100 FAT$=FAT$+STRING$(68-LEN(FA
T$),255)
1110 T=17:S=2:F$=""
1120 GOSUB 1480
1130 GDSUB 1810:DSKD$1,17,2,FAT$
,"":GOSUB 1790
1140 FOR I=3 TO 11
1150 GOSUB 1810
1160 DSKO$1,17,1,A$(I-1),B$(I-1)
1170 GDSUB 1790
1180 NEXT I
1190 GOTO 50
1200 CLS
1210 FC=0
1220 A$="GRANULE TABLE: "
1230 GOSUB 1350
1240 GDSUB 1810
1250 DSKI$ DV,17,2,A$,B$
1260 GOSUB 1790
1270 PRINTSTRING$(32,"-");
1280 FOR X=1 TO 68
1290 PRINT USING" %% ";HEX$(ASC(
MID$(A$,X,1)));
1300 IF ASC(MID$(A$,X,1))=255 TH
EN FC=FC+1
1310 NEXT X
1320 PRINT:PRINT"  FREE:"FC:PRIN
TSTRING$(32,"-");
1330 PRINT"PRESS ANY KEY."
1340 IF INKEY$>"" THEN 60 ELSE 1
340
1350 X1=0:X2=31
1360 A=LEN(A$)
1370 B=FIX(A/2)
1380 B$=LEFT$(A$,B)
1390 C$=(MID$(A$,B+1,A-B+1))
1400 X2=X2-LEN(C$)
1410 IF X1>=1 THEN PRINT@X1-1,"
";
1420 PRINT@X1,B$;
1430 PRINT@X2,C$
1440 X2=X2-1
1450 X1=X1+1
1460 IF X2=X1+(B-2) THEN 1470 EL
SE 1410
1470 PRINT@32,"";:RETURN
1480 '
1490 PRINT@171, USING"%
```

```
%";F$;
1500 PRINT@204, USING"%
     %";OP$;
1510 PRINT@238,"";:1F T=-1 OR S=
-1 THEN PRINT"      ";:GOTO 1520
 ELSE PRINT USING"## ##";T,S;
1520 PRINT@264,"";:IF G=-1 THEN
PRINT"   ";:GOTO 1530 ELSE PRINT
G;
1530 PRINT@293,"";:IF D=-1 THEN
PRINT"    ";:GOTO1540 ELSE PRINT
D;
1540 RETURN
1550 '
1560 ST=FIX(G/2)
1570 IF G>33 THEN ST=ST+1
1580 IF G/2=FIX(G/2) THEN SS=1:M
=0  ELSE SS=10:M=9
1590 ES=SS+8
1600 FOR Z=SS TO ES
1610 S=Z:T=ST:GOSUB 1480
1620 GOSUB 1810
1630 DSKI$D,ST,Z,C$(Z-M),D$(Z-M)
1640 GOSUB 1790
1650 NEXT Z
1660 RETURN
1670 '
1680 ST=FIX(G/2)
1690 IF G>33 THEN ST=ST+1
1700 IF G/2=FIX(G/2) THEN SS=1:M
=0 ELSE SS=10:M=9
1710 ES=SS+8
1720 FOR Z=SS TO ES
1730 S=Z:T=ST:GOSUB 1480
1740 GOSUB 1810
1750 DSKO$D,ST,Z,C$(Z-M),D$(Z-M)
1760 GOSUB 1790
1770 NEXT Z
1780 RETURN
1790 IF PEEK(&HFFFF)=27 THEN POK
E 65497,0 ELSE POKE 65495,0
1800 RETURN
1810 IF PEEK(&HFFFF)=27 THEN POK
E 65496,0 ELSE POKE 65494,0
1820 RETURN
```



Pressing the Reset button on the rear of the CoCo 3 does not clear the computer's memory. Yet turning the machine off and on again causes undue wear and tear. To completely reset the CoCo 3, erasing any programs in its memory, enter

```
POKE113,0:EXEC &HBC1B
```



TIM KIENTZLE

When is an Interpreter Better?

Many programmers routinely dismiss BASIC for a reason that has nothing to do with the language itself: Typically BASIC is implemented on microcomputers as a fairly simple interpreter, and it has thus earned a reputation as a slow language even though BASIC compilers can be used to create programs that run just as fast as their counterparts in other languages. Curiously, other interpreted languages have not earned this reputation. PostScript, APL, Smalltalk and Forth are all typically interpreted (in some fashion), but none of these is considered notoriously slow as is BASIC. To understand the difference, let's go back to the early days of computers and consider the controversy that once surrounded subroutines.

At one time all programming was done in machine code for computers that were puny by today's standards. Programmers stretched every bit of speed and memory efficiency by carefully rearranging and combining operations to take best advantage of whatever partial routines might already be available. Eventually a trick was discovered that allowed programs to have only one copy of certain routines — this is what we now call subroutines. The drawback was that it takes time to call a subroutine and return from it, and many programmers thought this additional time would result in unduly slow programs. However, they discovered that in a typical computation, almost all the time required was spent performing the instructions within the subroutine, and that the time to call and return from the subroutine made the program only slightly slower. It was clear that the memory savings of using subroutines far outweighed the slight additional time needed for the program to run, and the technique became common. Eventually users began loading collections of widely-used subroutines into the machine with every program, and these collections of subroutines became what we now call operating systems.

Although few people today would question the value of a subroutine, almost exactly the same situation occurs with an interpreted language. Each statement of the program being interpreted is really just a subroutine that results in a subroutine call within the interpreter. In this sense, the only difference between an interpreter and a compiler is that an interpreter figures out which subroutine to call as it reads each line, whereas a compiler figures this out once, and the compiled program simply calls the subroutines. What makes the interpreter slower is that it takes time to figure out which subroutine to call. If this time is a significant percentage of the total time, the interpreted version of the program is much slower. On the other hand, if the interpreter spends most of its time *in* the subroutines (i.e., actually doing the work), then the interpreted and compiled programs run at about the same speed.

The time needed to determine which subroutine to call is often referred to as the *interpretation overhead*. In a language like BASIC, a typical statement might cause two numbers to be added and stored in a variable. Since adding and moving numbers is very simple, the interpretation overhead does tend to take most of the time. In APL, a typical statement might cause a matrix to be inverted. Since inverting a matrix takes a very long time compared to the interpretation overhead, interpreted APL runs very nearly as fast as if it were compiled.

So, now we see that an interpreter can be very fast when the basic commands of the language perform very complex tasks. In PostScript, a single command can result in a very sophisticated (and time-consuming) graphics operation. This means that when selecting a language for writing a program, we should pay attention to how well the fundamental operations of the language match our job. Color BASIC, for example, does fairly well when the program emphasizes string, floating-point, and certain types of graphics operations. These are all relatively time-consuming operations that can be accomplished with only a few statements. BASIC does relatively poorly, however, when interpreting a program that performs extensive memory operations, since those are fairly simple operations.

As we've seen, an interpreted language need not be significantly slower than a compiled one. In fact, interpreted languages have advantages. Compiled programs are typically larger than their original source code, and interpreters usually use less space for storing programs. It is also easier to make interpreters work interactively, which makes it easier to debug and test programs. Finally, compilation itself can be time-consuming, so interpreters are often preferred if the resulting program is going to be run only a few times, as is the case with PostScript.

Even for languages that lack powerful fundamental operations, we shouldn't completely dismiss interpreters since major advances are being made that allow interpreters to run much faster. Forth and Smalltalk usually perform part of the interpretation once, storing some of the useful information. This is sometimes called *pseudo compiling*. This kind of technique is being pushed to the limit by companies writing *emulation programs*. Emulators are interpreters that interpret the machine code of another machine. For example, emulators have been developed that run MS-DOS software on Macintosh, Unix, Atari and other computers. Since machine instructions perform very simple operations, emulators are usually the slowest kind of interpreter. Methods being developed now to make emulators usably fast will probably be used someday to help interpreters of BASIC and other languages run more quickly. Indeed, it seems certain that interpreters will be more and more important as computer technology improves.

Tim Kientzle is currently pursuing a doctorate in mathematics at the University of California at Berkeley. He is the author of V-Term *and has worked with the Color Computer since 1982.*

Safari from Cover

number of cards remaining in the upper portion is displayed and you are asked if you want to play again. Press Y or N accordingly.

To quit the game in progress, press Q. You are asked if you would like to play again. This is a good way to start a new game if you don't like the cards dealt to you.

Darn It also supports an "oops" function. If you make a wrong move, press O and the game undoes the last move. You can press O repeatedly to go as far back in the current game as you want. This can be useful if you want to attempt a different strategy from a previous point. The Oops function works even after Darn It displays the No More Moves message.

Darn It is a very challenging game (perhaps this is why it so addictive). To make the game a little easier, you can eliminate the rule by which no card can be played on a King. Do this by deleting (or masking with REM statements) lines 1130 and 1320. Good luck!

Kenneth Reighard, Jr., is studying computer science and engineering at the University of Toledo, where he is a member of Triangle Fraternity. He enjoys programming, reading mystery novels and participating in sports. Ken may be contacted at 3355 Dorr St., Toledo, OH 43607, (419) 531-8149. Please include an SASE when requesting a reply.

CoCo 3

The Listing: DARN11

```
1 'DARN 1T
2 'BY KENNETH REIGHARD, JR.
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
100 DIM U(52), C(52), P(52), G(6
,6), S(15), R(6), O(52)
105 CLS:INPUT"MONITOR (R/C)";O$:
IF O$="R" THEN MN=1 ELSE IF O$=
"C" THEN MN=0 ELSE 105
107 CLS
110 POKE 65497,0
115 ON BRK GOTO 180
119 GOSUB 1500 'BLANK SCREEN
120 GOSUB 1800 'DRAW CARDS
123 GOSUB 2400 'DRAW TITLES
125 GOSUB 1400 'RESET COLORS
127 GOSUB 2500 'PLAY MUSIC
130 GOSUB 200 'SHUFFLE CARDS
140 GOSUB 300 'DEAL CARD ARRAYS
150 GOSUB 400 'SET UP SCREEN
160 GOSUB 500 'PLAY GAME
165 HCOLOR 7:HPRINT(11,15),"Play
 Again (Y/N)?"
170 O$=INKEY$:IFO$="Y" THEN HCLS
:GOTO 130 ELSE IF O$="N" THEN 18
0 ELSE IF O$="O" THEN GOSUB 2600
:HLINE(72,112)-(264,128),PRESET,
BF:GOTO 160 ELSE 170
180 GOSUB 1400:POKE 65496,0:CLS:
END
199 'SHUFFLE CARDS
200 FOR X=1 TO 52:U(X)=0:NEXT X
210 X=RND(-TIMER)
220 FOR X=1 TO 52
230 Y=RND(52):IF U(Y)<>0 THEN 23
0
240 U(Y)=I:C(X)=Y
245 NEXT X
250 RETURN
299 'SET UP CARD ARRAYS
300 P=1
310 FOR X=1 TO 6:FOR Y=1 TO 6
320 G(X,Y)=C(P)
325 P=P+1
330 NEXT Y,X
340 FOR X=1 TO 15
350 S(X)=C(X+36)
360 NEXT X
370 P(1)=C(52)
372 FOR X=1 TO 6:R(X)=6:NEXT X
374 Z=15:O=1:CL=36:O=0
380 RETURN
399 'SET UP SCREEN
400 HCOLOR 4,8:HCLS
410 FOR X=1 TO 6:FOR Y=1 TO 6
420 C=G(X,Y):GOSUB 700
430 NEXT Y, X
440 X=2:Y=17:C=78:GOSUB 700
450 X=5:Y=17:C=P(1):GOSUB 700
455 GOSUB 900
460 RETURN
499 'PLAY GAME
500 V=1
510 H=V*40+17:HDRAW"C2BM=H;,I12X
A$;"
520 Q$=INKEY$:IF Q$="" THEN 520
530 HDRAW"CBBM=H;,112XA$;"
540 IF Q$=CHR$(8) THEN V=V-1:IF
V<1 THEN V=6
550 IF Q$=CHR$(9) THEN V=V+1:IF
V>6 THEN V=1
560 IF Q$=CHR$(32) THEN GOSUB 10
00
570 IF Q$=CHR$(13) THEN GOSUB 11
00
580 IF Q$="Q" THEN 650
585 IF Q$="O" THEN GOSUB 2600
590 IF CL=0 THEN GOSUB 2200:GOTO
 650
592 IF Z=0 THEN GOSUB I300:IF NO
T(CM) THEN GOSUB 2300:GOTO 650
600 GOTO 510
650 RETURN
699 'PUT CARD ON SCREEN
700 HPUT(X*40-1,Y*8-2)-(X*40+33,
Y*8+52),(INT(C/14)+1)
710 CC=C
712 IF CC>52 THEN 750
715 IF CC<27 THEN HCOLOR 3 ELSE
HCOLOR 8
720 GOSUB 800
730 IF CC=1 THEN C$="A" ELSE IF
CC<10 THEN C$=RIGHT$(STR$(CC),1)
 ELSE IF CC=10 THEN C$="10" ELSE
 IF CC=11 THEN C$="J" ELSE IF CC
=12 THEN C$="Q" ELSE C$="K"
740 HPRINT(X*5,Y),C$
750 RETURN
799 'GET CARD VALUE
800 IF CC>13 THEN CC=CC-13:GOTO
800
810 RETURN
899 'PRINT CARDS LEFT IN DECK
900 HCOLOR5:HLINE(88,176)-(104,1
84),PSET,BF
910 HCOLOR7:IF Z>9 THEN HPRINT(1
0,22),Z ELSE HPRINT(11,22),"0"+R
IGHT$(STR$(Z),1)
920 RETURN
999 'GET CARD OFF DECK
1000 IF Z=0 THEN 1030
1005 O=O+1:P(O)=S(Z)
1010 X=5;Y=17;C=P(O):GOSUB 700
1020 Z=Z-1:IF Z=0 THEN X=2:Y=17:
C=65:GOSUB 700 ELSE GOSUB 900
1025 O=O+1:O(O)=0
1027 PLAY"O3L100C"
1030 RETURN
1099 'PLAY CARD FROM GRID
1100 IF R(V)=0 THEN 1210
1110 CC=P(Q):GOSUB 800:QQ=CC
1120 CC=G(V,R(V)):GOSUB 800
1130 IF QQ=13 THEN GOSUB 1600:GO
TO 1210
1140 IF NOT(CC=QQ+1 OR CC=QQ-1)
THEN GOSUB 1700:GOTO 1210
1150 X=V:Y=R(V):C=65:GOSUB 700
1160 Q=Q+1:P(Q)=G(V,R(V)):R(V)=R
(V)-1
1170 IF R(V)>0 THEN X=V:Y=R(V):C
=G(V,R(V)):GOSUB 700
1180 X=5:Y=17:C=P(O):GOSUB 700
1185 O=O+1:O(O)=V
1190 PLAY"O2L100ABDC"
1200 CL=CL-1
1210 RETURN
1299 'CHECK FOR LEGAL MOVES
1300 CM=0
1310 CC=P(O):GOSUB 800:QQ=CC
1320 IF QQ=13 THEN 1370
1330 FOR X=1 TO 6
1335 IF R(X)=0 THEN I360
1340 CC=G(X,R(X)):GOSUB B00
1350 IF CC=QQ+1 OR CC=QQ-1 THEN
CM=1
1360 NEXT X
1370 RETURN
1399 'SET COLORS
1400 IF MN THEN RGB ELSE CMP
1410 RETURN
1499 'SET COLORS TO BLACK
1500 FOR X=0 TO 8:PALETTE X,0:NE
XT X
1510 RETURN
1599 'PLAY ON KING MESSAGE
1600 HCOLOR 3:HPRINT(6,15),"Can'
t play a card on a King."
I610 PLAY"L100D2ECECECECECECFC
P8"
1620 HLINE(48,120)-(272,I28),PRE
SET,BF
I630 RETURN
1699 'ILLEGAL MOVE MESSAGE
1700 HCOLOR 3:HPRINT(13,15),"Ill
egal Move."
1710 PLAY"L100O1CACACACACACACA
CP8"
1720 HLINE(104,120)-(208,128),PR
ESET,BF
1730 RETURN
1799 'DRAW CARD TEMPLATES
1800 HCOLOR 8,4:HSCREEN 2
1810 FOR X=I TO 6
1820 H8UFF X,993
1830 HLINE(6,7)-(42,61),PSET,B
1840 ON X GOSUB 1900,1930,1960,1
990,2020,2040
1850 HGET(6,7)-(40,6I),X
1860 HCLS
1870 NEXT X
1880 A$="U10NF5G5"
1890 RETURN
1900 HDRAW"C3BM23,38H5U2ER2F2E2R
2FD2G5C8"
1910 HPAINT(22,36),3,3
1920 RETURN
1930 HORAW"C3BM23,38H5E5F5G5CB"
1940 HPAINT(22,36),3,3
1950 RFTURN
1960 HDRAW"8M23,38LU3G2H2E2R2H2E
3F3G2R2F2G2H2D3L"
1970 HPAINT(23,36),8,8
1980 RETURN
1990 HDRAW"BM23,38LU3G2L2U4E5F5D
4L2H2D3L"
2000 HPAINT(23,36),8,8
2010 RETURN
2020 HPAINT(22,36),8,B
2030 RETURN
2040 HPAINT(23,42),5,B
2050 FOR Y=3 TO I5 STEP 3
2060 HCIRCLE(24,42),Y,0,1,.5,0
2070 NEXT Y
2080 HCOLOR 0
2090 HLINE(8,43)-(3B,43),PSET
2100 FOR Y=0 TO 3:HPAINT(24-(Y*3
+4),42),Y,0:HCIRCLE(24,42),Y*3+3
,Y,1,.5,0:NEXT Y
2110 HCIRCLE(24,42),15,5
2120 HCOLOR 5:HLINE(8,43)-(38,43
),PSET
2130 HC1RCLE(24,20),4,I:HPAINT(2
3,20),1,1
2140 HCOLOR 7:FOR Y=1 TO 2:HLINE
(6+Y,7+Y)-(42-Y,61-Y),PSET,B:NEX
T Y
2150 RETURN
2199 'WIN MESSAGE
2200 HCOLOR 1:HPRINT(I1,14),"We
Have a Winner!!"
2210 PLAY"O2L8G#GF#GG#GF#P8D#DC#
OO#DC#PBC#D#F#P8C#D#F#PBF#GG#A#B
P255BP255B"
2220 RETURN
2299 'LOSE MESSAGE
2300 HCOLOR 3:HPRINT(9,14),"No m
oves,"+STR$(CL)+" cards left."
23I0 PLAY"O1L4EDL2C"
2320 RETURN
2399 'TITLE SCREEN
2400 HCOLOR 4,B:HCLS
2410 Q$="C3O5ØR1ØE1ØU3ØH1ØL1ØBR4
5BDIØC2NFIØG1ØD1ØNO3ØR2ØNU1ØD3ØB
R15BU4ØCIND5ØRIØF1ØG1ØNL1ØM+1Ø,+
3ØBR15BU4ØC6ND5ØM+2Ø,+5ØU5ØBR35B
U1ØC7R1ØNR1ØD5ØNL1ØR1ØBR15BU6ØC5
R1ØNR1ØD5ØBR25BU6ØCØD4ØBD5O5"
2420 FOR X=1 TO 5
2430 H=40+X;V=I0+X;HORAW"BM=H;,=
V;XQ$;"
2440 NEXT X
2450 HCOLOR 4:HPRINT(14,13),"A C
ard Puzzle"
2460 HPRINT(8,18),"By Kenneth Re
ighard, Jr."
2470 HPRINT(13,23),"Copyright 19
92"
2480 RETURN
2499 'THEME MUSIC
2500 FOR X=1 TO 17
2510 PLAY"O2T=X;L4F#GG#"
2520 NEXT
2530 PLAY"T2L2G"
2540 RETURN
2599 'OOPS ROUTINE
2600 IF O=0 THEN 2650
2610 IF O(O)=0 THEN Z=Z+1:O=O-1:
IF Z=1 THEN X=2:Y=17:C=78:GOSUB
700:GOSUB 900:GOTO 2630 ELSE GOS
UB 900:GOTO 2630
2620 R(O(O))=R(O(O))+1:O=O-1:X=O
(O):Y=R(O(O)):C=G(O(O),R(O(O))):
GOSUB 700:CL=CL+1
2630 X=5;Y=17;C=P(O):GOSUB 700
2640 O=O-1
2645 PLAY"L10001BD"
2650 RETURN
```

Prevent Monitor Burn-in

by Frank D'Urso

If you've ever written a program that uses INKEY$ (or any similar procedure) to pause for user input, this utility is for you. While the computer is waiting, your monitor or television is burning ever onward. A screen saver is designed to either blank the screen or display moving graphics in an attempt to keep the screen image from being burned onto the inside of the monitor. My version, which for the sake of simplicity I call Screen Saver, gives you moving graphics along with musical tones. Even if you decide not to use it to save your monitor, it's fun to watch the program do its work.

Screen Saver is designed to relieve your monitor as well as your eyes and ears. Updating previous graphics screen pleas, I've linked sounds to match the x and y coordinates of a series of lines that move in a "three-dimensional" space. The lines begin at the top of the screen and flutter, twist and bounce around the monitor across a black background. After every 400 or so lines appear, the color used to draw them is changed.

Screen Saver can be run as a stand-alone program, or it can be incorporated as a subroutine from INPUT-type statements in your BASIC creations. To see the program in action, first enter it as shown and save it to tape or disk. Then run it and enjoy.

The program uses the high-speed poke (Line 80) for operation. If you press BREAK, Line 80 has also set up a "trap" to send the program to Line 390 where normal speed is restored. If the program crashes, enter either POKE 65496,0 or RUN 390 to slow the computer. Never try to save or load programs while the CoCo is in the high-speed mode.

Screen Saver is fun to watch and listen to. I hope it adds to your enjoyment and becomes a useful part of your own programming efforts.

Frank D'Urso has a degree in communications from Northeastern University and is currently pursuing a masters in government at Harvard. He has worked in advertising, and he enjoys art, music and computer games. Frank may be contacted at 38 Westford St., Saugus, MA 01906, (617) 666-2137. Please include an SASE when requesting a reply.

CoCo 3

The Listing: SSAVER

```
1 'SCREEN SAVER
2 'BY FRANK D'URSO
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
70 A=8
80 PALETTE RGB:POKE 65497,0:HSCR
EEN2:HCLS4:ON BRK GOTO 390
90 X=RND(2)
100 D=RND(3)
110 B=RND(3)
120 Q=RND(3)
130 Z=RND(7)
140 Z=Z+1
150 IF Z>8 THEN Z=Z*0
160 FOR Y=1 TO 400
170 B=B+1:S=S+.1:D=D+Q
180 IF B>250 THEN B=B-3
190 Y=Y+1
200 HCOLOR Z
210 IF B>250 THEN B=B+1
220 IF D>180 THEN Q=Q-2
230 IF D<5 THEN Q=Q+2
240 IF B<5 THEN B=B+2
250 X=(SIN(S)*160)+160
260 IF X<1 THEN X=X+2
270 IF D<1 THEN D=D+2
280 IF B<1 THEN B=B+2
290 HLINE(X,D)-(0,B),PSET
300 M=250-D
310 N=250-B
320 IF M<1 THEN M=M+10
330 IF N<1 THEN N=N+10
340 SOUND M,1
350 SOUND N,1
360 NEXT Y
370 X=X*0:B=B/20
380 GOTO140
390 POKE 65496,0:CLS0:END
```

Product Review

The *MM/1 Technical Reference Manual*

The MM/1 Technical Reference Manual consists of ninety-one 6-by-8½ pages in a looseleaf binder. It was written by Mark Griffith and Carl Kreider, two programmers with a great deal of experience (and no little renown) in the OS-9 world. As a technical guide to the MM/1 68000-based computer, does the manual live up to the standards one expects of this team? Well, yes . . . and no. Let's look at the no side first.

The MM/1 Technical Reference Manual lacks many of the components often considered essential in a reference intended for technicians who are expected to repair or upgrade a modern microcomputer. There are no complete parts lists, no complete schematics and no diagnostic code reference (though a call to Interactive Media Systems confirmed that the ROMs do supply diagnostic codes to the technician).

From a production aspect, there are a few irritating problems. Pages iii and iv in the Table of Contents cover only seven of 15 chapters. Pages i and ii do not exist in the copy I received, leaving me to wonder just what was left out. In addition, some pages exhibit reproduction problems, which would ordinarily be simply annoying. However, the edges of some of the partial schematics wander close to the edges of the pages, some labels have been "truncated" (a nice word for cut off). In certain places, cross references and tables are not where the book indicates. Finally, some of the information provided in the MM/1 Technical Reference Manual is out of date, left behind by rapid changes in the drivers and descriptors during this powerhouse machine's infancy.

What is in the MM/1 Technical Reference Manual? The manual contains a lot of information useful to programmers. For instance, Chapter 3 provides a discussion of the MM/1 memory maps. That's right, maps is plural. The MM/1 has separate memory maps for the one-, three- and nine-megabyte systems.

Chapter 12 supplies a fairly comprehensive table of jumper settings. The only problem here is that, while it tells you which jumper does what, the manual fails to describe which features the various jumper settings select, leaving this as an exercise for the technician.

Chapters 6 and 7 discuss the MM/1 SCSI system, and the use of SCSI hard drives, respectively. Chapter 6 is not a complete guide to programming a SCSI driver, but it does provide enough information to give the user an understanding of the MM/1 SCSI driver system. Chapter 7 is a very informative discussion of how to add new hard drives to the MM/1. Floppy drives are thoroughly discussed in Chapter 8, as is the use of dmode to change disk formats.

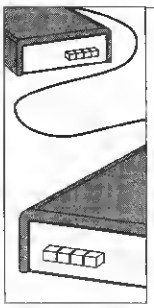
In my opinion, the meat of the MM/1 Technical Reference Manual is in chapters 9 and 10, which explain the characteristics of the MM/1's I/O chips. The information given here does not represent a complete guide to the chips in general (the text explains that this is due to the increased complexity of the LSI devices used in the MM/1). Rather, these chapters outline those features specific to the MM/1. Here you will find the addresses of those registers and ports actually used in the MM/1 system. These chapters offer some useful hints on such items as how to determine the status of the CD line and how to toggle keyboard and floppy-drive interrupts.

While chapters 9 and 10 do not disclose complete technical data on every chip in the MM/1, Chapter 15 supplies you with addresses where you can obtain complete manuals for each chip. My online sources tell me the volume of data available from the companies listed in Chapter 15 (especially Signetics) is nothing short of phenomenal!

How does the MM/1 Technical Reference Manual compare with other technical manuals? To answer this question, I asked several technicians for their opinions. MS-DOS support personnel raved about the layout of the manual and called it "Superb . . ." an outstanding manual that is easy to read." Technicians who work with peripheral equipment and Unix systems were less happy, stating that the lack of schematics, parts lists and diagnostic-code references made the manual more suitable for programmers than for hardware repair sites.

Even for the price of $49.95, the MM/1 Technical Reference Manual forms a priceless source of data for software developers striving to carry on the traditions founded by the CoCo Community. (Interactive Media Systems, 1840 Biltmore St. NW, Washington, D.C. 20009; $49.95.)

— David M. Graham



INTERCOM

Pen pals

✍ I have a 128K CoCo 3 with an FD-502 disk drive, a DMP-106 printer and a color TV. I would like to hear from pen pals between the ages of 8 and 12, but I will write to people of all ages.

Kevin Smith
1958 Washington Avenue
Portland, ME 04103

CoCo clubs

CALIFORNIA

☎ StG Net West, Alan Sheltra, P.O. Box 38713, Hollywood, 90038, (818) 761-4135, BBS (818) 761-4721

COLORADO

☎ Colorado Springs Color Computer Club, Bud Ward, 1118 Claiborne Road, Colorado Springs, 80906-5513, (719) 392-8268

CONNECTICUT

☎ Connecticut CoConut Connection, Charles Joseph Scanlon, 2 Eagle Lane, Simsbury, 06070, (203) 657-8373

FLORIDA

☎ The Color Computer 3 Users Group, Tom Batchelder, 6042 Syrcle Ave., Milton, 32570, (904) 623-4405

GEORGIA

☎ Atlanta Computer Society, Inc., Alan R. Dages, 4290 Bells Ferry Road Suite 10639, Kennesaw, 30144, (404) 469-5111 voice, (404) 636-2991 modem

IDAHO

☎ Snake River Color Computer Club, Emil Franklin, 1750 Carmel Drive, Idaho Falls, 83403, (208) 522-0220

ILLINOIS

☎ Cook County Color Computer Club, Howard Luckey, 10 McCarthy Rd., Park Forest, 60466-2122, (708) 747-0117

☎ Motorola Micro Computer Club, Steve Adler, 1301 East Algonquin Rd., Shaumburg, 60196, (708) 576-3044

IOWA

☎ Metro Area Color Computer Club, Joe Cavallaro, 2425 Ave A, Co. Bluffs, 51501, (712) 322-2438

☎ Mid Iowa & Country CoCo, Terry Simons, 1328 48th Street, Des Moines, 50311, (515) 279-2576

KENTUCKY

☎ Hardin County Color Computer Club, Paul Urbahns, 2887 Republic Ave., Radcliff, 40160, (502) 351-4757

LOUISIANA

☎ The CoCo SIG, Christopher Mayeux, 20 Gibbs Drive, Chalmette, 70043, (504) 277-6880 voice, (504) 277-5135 modem

MARYLAND

☎ Arkade, John M. Beck, 3513 Terrace Drive #D, Suitland, 20746, (301) 423-8418

MASSACHUSETTS

☎ NorthEast CoCo Club, Jose Joubert, 440 North Ave., Bldg. 9 #210, Haverhill, 01830, (508) 521-0164

MICHIGAN

☎ Color Computer Owners Group, Bernard A. Patton, 388 Emmons Blvd., Wyandote, 48192, (313) 283-2474

☎ Greater Lansing Color Computer Users Group, E. Dale Knepper, P.O. Box 14114, Lansing, 48901, (517) 626-6917

MISSISSIPPI

☎ Mississippi OS-9 User Group, Boisy G. Pitre, Southern Station, Box 8455, Hattiesburg, 39406-8455, (601) 266-2807

MISSOURI

☎ CoCoNuts User Group, Clyde Lloyd, 2116 N. Columbia, Springfield, 65803, (417) 866-8738

☎ KC CoCo, Gay Crawford, P.O. Box 520084, Independence, 64052, (913) 764-9413

NEBRASKA

☎ Bruce Gerst c/o Metro Area CoCo Club, P.O. Box 3422, Omaha, 68103

NEW YORK

☎ Erie County Color Computer Club, John A. Lombardo, 57 Chapel Ave., Cheektowaga, 14225

NORTH CAROLINA

☎ Raleigh CoCo Club, P.O. Box 10632, Raleigh, 27605, (919) 878-3865

☎ The Tandy Color Computer Users of Charlotte, Eric Stringer, 1022 Noles Dr., Mt. Holly, 28120

OHIO

☎ The Greater Toledo Color Computer Club, Bill Espeu, 1319 North St., Bowling Green, 43402, (419) 471-9444

☎ Tri-County Computer Users Group, Ron Potter, 10914 Oliver Road, Cleveland, 44111, (216) 476-2687

PENNSYLVANIA

☎ Cumberland Valley Users Group, Thomas Martin, 9085 Newhurg Road, Newburg, 17240, (717) 423-5525

RHODE ISLAND

☎ New England "CoCoNuts" Color Computer Club, Arthur J. Mendonca, P.O. Box 28106 North Station, Providence, 02908, (401) 272-5096 (Sig3)

SOUTH CAROLINA

☎ Spartanburg CoCo Club, Jesse W. Parris, 152 Bon Air Ave., Spartanburg, 29303, (803) 573-9881

SOUTH DAKOTA

☎ Empire Area Color Computer Users Group of South Dakota, Carl Holt, P.O. Box 395, Brandon, 57005, (605) 582-3862

TEXAS

☎ The Codis CoCo Symphony, William C. Garretson, 2902 Harvard St., Irving, 75062, (214) 570-0823

UTAH

☎ Salt City CoCo Club, L. Todd Knudsen, 6357 S. Lotus Way, West Jordan, 84084, (801) 968-8668

WASHINGTON

☎ Bellingham OS-9 Users Group, Rodger Alexander, 3404 Illinois Lane, Bellingham, 98226, (206) 734-5806

☎ Port O' CoCo, Donald Zimmerman, 3046 Banner Rd. SE, Port Orchard, 98366-8810, (206) 871-6535

AUSTRALIA

☎ Australian National OS-9 Users Group, Gordon Bentzen, C/- 8 Odin Street, Sunnybank, Queensland, 4109, (07) 344-3881

☎ Brisbane Southwest Colour Computer Users Group, Bob Devries, 21 Virgo St., Inala, Queensland, 4077, (07) 372-7816

CANADA

☎ Club d'Oridinateur Couleur du Quebec Inc., 8000 Metropolitain est, Anjou, Quebec, H1K 1A1, (514) 354-4941

GERMANY

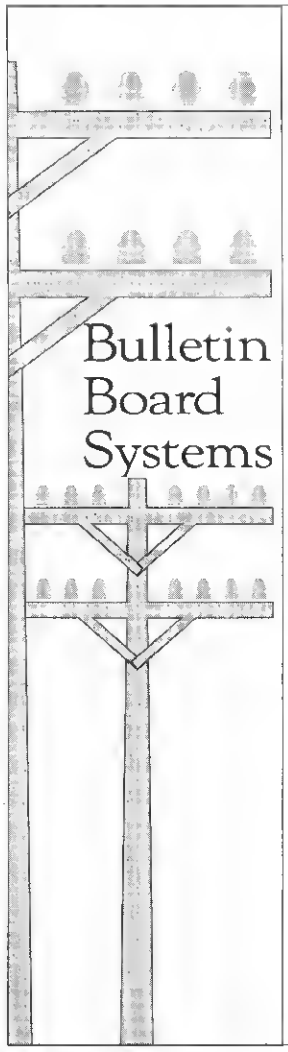
☎ OS-9 Users Group in Europe, Burghard Kinzel, Leipziger Ring 22A, 5042 ERFTSTADT, +49-2235-41069, (OS-9/6809)

THE NETHERLANDS

☎ European OS-9 User Group, Peter Tutelaers, Strijperstraat 50A, 5595 GD Leende, s88405777@hsepml.hse.nl, +31-4906-1971, (OSK)

PUERTO RICO

☎ Puerto Rico Color Computer Club, Luis R. Martinez, P.O. Box 2072, Guaynabo, 00657-7004, (809) 799-8217 or (809) 728-2314



BBS's

State/City	BBS Name	Access Number	Parameters (Speed-Parity-Word Bits-Stop Bit)	SysOp
Arkansas				
Sheridan	The Grant County BBS	(501) 942-4047	300/1200/2400-N-8-1	Eddie Gilmore
Califurnia				
Hollywood	Zug's Cavern BBS	(213) 461-7948	300/1200/2400-N-8-1	Alan Sheltra
Connecticut				
Manchester	Silk City BBS	(203) 649-9057	300/1200/2400-N-8-1	Darren Kindberg
Waterbury	Applause BBS	(203) 754-9598	300/1200/2400-N-8-1	Carmen Izzi, Jr.
Hawaii				
Ft. Shafter	CoCo'Nuts BBS Service	(808) 845-7054	300/1200/2400-N-8-1	Tommie Taylor
Idaho				
Idaho Falls	Snake River Computer Club BBS[1]	(208) 523-3796	300/1200-N-8-1	Jon Gould
Illinois				
Carpentersville	The Pinball Haven BBS	(708) 428-8445	300/1200/2400-N-8-1	Jeffrey R. Chapin
Elmhurst	Glenside's Cup of CoCo BBS	(708) 428-0436	300/1200/2400-N-8-1	Tony Poliraza
Kentucky				
Elkhorn City	Cross-N-Crown BBS	(606) 754-9420	300/2400-N-8-1	Tim McIntush
Michigan				
Manistee	Crystal Palace	(616) 723-0146	1200/2400-N-8-1	Nelson Howard
Mississippi				
Hattiesburg	The OS-9 Zone[2]	(601) 266-2807	300/1200/2400-N-8-1	Boisy G. Pitre
New York				
Erie County CoCo Club		(716) 649-1368	300/1200/2400-N-8-1	Wayne Mullen
Wappingers Falls	The Dutchess CoCo	(914) 838-1261	300/1200/2400-N-8-1	Chris Serino
North Carolina				
Wilmington	Bill's Board	(910) 395-4366	300/1200/2400-N-8-1	Bill Medcalf
North Dakota				
Minot AFB	The 9-Line BBS	(701) 727-6826	300/1200-N-8-1	David Hensley
Ohio				
Columbus	Springwood BBS	(614) 228-7371	300/1200/2400-N-8-1	Edward Langenback
Pennsylvania				
Conshohocken	Charlie's Help Line	(215) 825-3226	300/1200-N-8-1 or N-7-1	Charles DiMartino
Rhode Island				
Central Falls	The Weather Connection II BBS	(401) 728-8709	300/1200/2400-N-8-1	Eric Chew
Virginia				
Fall Mills	Clem's Corner BBS[3]	(703) 322-4053	300/1200-N-8-1	Richard Douglas Bailey
Washington				
Fircrest	OS-9 Tacoma	(206) 566-8857	300/1200/2400-N-8-1	Chris Johnson
Wisconsin				
Marinette	Phoenix Interstate Data Systems[4]	(715) 732-1036	300/1200/2400/9600-N-8-1	Joe Boburka
Canada				
Twillingate, NF,	ColorNET BBS	(709) 884-2176	300-N-8-1	Jason Woodford
Windsor, Ontario	Color Connection	(519) 948-1879	300/1200-N-8-1	Cory Richert

Notes:
[1]Snake River Computer Club BBS supports all types of computers.
[2]The OS-9 Zone is up from 10 p.m. to 6 a.m. seven days a week.
[3]Clem's Corner BBS is up from 6 p.m. to 11 p.m. seven days a week.
[4]Phoenix Interstate Data Systems has a .75/hr charge for premium services, paid in advance.

MARTY GOODMAN

End Packing

I recently was examining a favorite CoCo utility called BOOT.BAS, *which displays a directory on the screen and allows me to use the arrow keys to select a program. It then runs that program when I press ENTER. At first glance the program is simple, but closer study reveals that a machine-language program is hidden inside the* BASIC *program. It took me a lot of sleuthing to figure out how the machine-language program was hidden, but I eventually figured it out: It is located at the end of the* BASIC *program. The author stored it just after the last statement in the* BASIC *program, manually changed the end-of-*BASIC *program pointer (located at &H001C and &H001D) to point beyond the end of the machine-language program, then saved the whole program to disk. The result is a* BASIC *program that pulls into memory a machine-language program any time it was loaded. Do you know who wrote this program? Weren't you associated with it at one time?*

George Quellhoerst
Bainsville, Ohio

Yes, I was involved in distributing that program many years ago. It was written by my good friend Peter Ryan (N6LQV), author of *WEFAX*, *RTTY* and *Graphicom*. The technique used is what I call "end packing" since the machine-language program is packed between the end of the BASIC program and the end-of-program pointer. This is a convenient way to bring a machine-language routine into memory when it is associated with a BASIC program. There was never any intent to hide what was going on, but the technique is sufficiently tricky that it can appear cryptic if you are not familiar with it. Indeed, this technique was once employed by various CoCo software-protection schemes in an effort to slow those who would attempt to figure out the protection.

One curious thing about end-packed programs is that they often can't be uploaded or downloaded properly with Xmodem. A far more common approach to include a machine-language program with a BASIC program is to have the BASIC program poke the machine-language program into memory from data statements. This approach has the advantage of being amenable to transfers over a modem, and it is a lot easier to understand. However, it is slow and causes the machine-language program to take up more room than it would if it were end-packed.

Replacement Chips

Where can I get replacement chips for the 512K upgrade, the FD-502 disk controller (or older-model controllers), the Radio Shack Multi-Pak Interface, the Orion Telepak and the Burke & Burke CoCo-XT real-time clock?

Greg Morgan (AIRWOLF2)
Richmond, Virginia

The 512K upgrade board uses sixteen 41256 (120- or 150-ns) dynamic RAM chips. These are widely available for between 50 cents and a dollar apiece from most chip suppliers. (Microprocessors Unlimited in Beggs, Oklahoma, is often a good source of memory chips.)

Later-model CoCo disk controllers, including the FD-502, typically use either a 28-pin 1773 controller chip. Earlier controllers usually use a 1793, 5-volt-only, 40-pin controller chip. These have not been made for a long time and can be quite hard to locate. (I can't help you there.) All the other chips (except the 8-pin data separator in controllers using the 1793-chip) are generic small-scale logic chips and are available from any standard chip supplier, such as JDR in San Jose, California.

Like those in the controllers, all the chips in the 26-3024 Multi-Pak Interface are standard TTL logic chips. The newer, smaller 26-3124 model Multi-Pak Interface uses one 64-pin ASIC (Application Specific Integrated Circuit), which was custom made for Tandy and is, I suspect, no longer available. However, the rest of the chips in that unit are standard TTL logic chips. As a side note, in most cases of a dead Multi-Pak, the ASIC chip is not what is affected. Rather the buffer chips are what get fried. These buffer chips are 35-cent generic TTL chips, available from JDR and most other chip vendors.

The Orion Telepak and other RS-232 packs use the 6551A ACIA (which I believe is available from JDR) and one or two level-converter chips and logic chips. Some models of the Orion Telepak and the Tandy RS-232 Pak may have used a DC-to-DC voltage inverter to create +12 and -12-volt sources from a single +5-volt source. This module may be hard to find (at least in small quantities) since it is not a commonly used part. If the inverter does die (this is trivial to check: feed it +5 volts and see if it delivers the required +12 and -12 volts), you can power the level-converter chips directly from the +12 and -12-volt lines in the Multi-Pak Interface. Alternatively, use a MAX 232 or 233 level converter, both of which have internal DC-to-DC voltage-conversion circuitry. Of course, this would require extensive rewiring.

Consult Burke & Burke regarding a spare real-time clock chip. I believe the real-time clock uses a commonly available OKI brand real-time clock chip.

By the way, unless you have a background in electronics or a great deal of experience, random replacement of chips in a dead device is not likely to result in a fix. And if it does, it certainly isn't likely to happen in a timely or economic fashion.

Bit Banging

Is there a way to use the 4-pin printer port on the CoCo under OS-9 with a modem and an OS-9 telecommunications program?

Alain Pilon (APILON)
Brossard, Quebec
Canada

Because driving the 4-pin "bit-banger" serial port on the CoCo eats up so much processor time, other OS-9 tasks come to a screeching halt. There is available on Delphi a driver for the port that some say enables the computer to be used at 1200 bps with a modem (as long as you make a custom cable that feeds the receive-data line into an interrupt line on the port). Even then, operation is likely to be unreliable, and you won't be able to run much else than your terminal program while this driver is in use. So, while it is possible to use the bit-banger port for modem communications, it is not advisable. If you are using OS-9, you really need a hardware RS-232 pack for reliable OS-9 modem communication.

Which Hard Drive Is It?

I have a Quantum drive that bears a model number that appears to be

either "QS40" or "QS4C." I have been told the drive has a storage capacity of 35 megabytes. Can it be used with a Color Computer? What would I have to get to use it with the CoCo? I bought this drive during an auction at a local university.

Joe Villarreal (VILLAREAL)
Lubbock, Texas

A The drive you are talking about is almost certainly a Quantum Q-540 drive, which is listed in my references as a full-height, 5¼-inch, 36-megabyte MFM drive. It appears to be an extremeley ancient drive that does not have automatic head acceleration and deceleration, but instead steps the head at a constant rate causing exceedingly slow hard-drive access. Even if the drive were brand new, I'd recommend that you not use it. Knowing that it was used in a university environment, where it probably received heavy use, I make this recommendation a fairly strong one.

The cost of the actual hard drive is usually a small fraction of the total cost of a hard-drive system for the CoCo. With the RGB/Ken-Ton system, you'll need to purchase a host adaptor and either a SCSI drive (support is provided for a limited number of types) or a SCSI controller card with the proper ROM to drive an MFM drive. With the Burke & Burke system, you'll need the CoCo-XT adapter and an 8-bit PC-type hard-drive controller. Whichever route you take, you'll also need cables, a case and power supply, and appropriate driver software. The Burke & Burke system requires a Multi-Pak Interface or a highly modified Y cable is required. The RGB/Ken-Ton system is ready-to-run with a Y cable. Unless you are buying a new 80-meg or larger drive, the hard drive itself is going to account for much less than half the total cost of the system.

Televideo Terminals

Q *I have an old Televideo terminal but no documentation for it. I'm seeking help on what functions its various DIP switches perform, especially those that set the serial-port speed.*

Tony Reed (TONYREED)
Montreal, Quebec
Canada

A In the past I've used a CoCo running a terminal program and a null-modem cable to identify DIP-switch functions on unfamiliar terminals. With the two "terminals" linked, start varying the serial-port speed and other parameters on the CoCo until you are able to display characters from the terminal keyboard on the CoCo screen and vice versa. Then, *after carefully recording the positions of all the DIP switches*, alter the combinations one by one to see if the speed changes. Once you have determined which switches control the speed, alter CoCo's speed until you get readable characters again. With this approach, it usually takes little time to document all the settings for the terminal. With a little luck and a lot of trial and error, you can use a variant of this technique to determine control settings for the serial printer port that most such terminals include.

Feature-Rich or Feature-Bloated?

Q *Why are PC-compatible programs (such as terminal programs and word processors) so much bigger than their Color Computer 3 counterparts? For example, PC-compatible word processors are between one and five megabytes in size, compared to 40K CoCo programs. This is a pretty big size difference. What's going on?*

Charles A. Marlow (CHARLESAM)
Massapequa, New York

A There are several factors at play here. The MS-DOS market is feature-driven, and the most common form of competing in the market is to pack more features into a program, as opposed to making the core functions of the program function especially quickly or elegantly. Thus, MS-DOS programs are in general "feature-bloated" — they have far more options and functions (some useful, some not) than an equivalent CoCo product. Note that by the standards of the PC-compatible world, word processors for the CoCo are mere "text editors." And PC-compatible word processors are, by the standards of a few years ago, full-fledged desktop-publishing packages.

Another aspect to consider is that most MS-DOS software comes with literally hundreds of printer drivers and many auxilliary programs. For example, few (if any) MS-DOS word processors don't include a spelling checker and thesaurus. All of this contributes to the bulk of the package.

Much MS-DOS software is written in higher-level languages and compiled to machine code, resulting in much larger executable files. In contrast, higher-power CoCo applications are usually written in assembly language from the start. In one sense, CoCo programmers have to be more skilled in getting the most out of a machine than PC programmers, who can count on massively powerful hardware to make up for inefficient code. For example, I use *Professional Write* (an "also ran, beginner type" package) on my MS-DOS machine. This program takes about 30 seconds to change the margins on a 20-page document on a 12-MHz 286 computer. CoCo word processors handle the same job in a second or less, due to much tighter code.

Traditionally the biggest memory hogs in MS-DOS software are those programs that use graphic user interfaces. Massive amounts of memory must be used to store icons, fonts, etc. Thus *Windows* and applications for that environment are truly enormous.

Finally, there is one principle that affects all but the best programmers: Programmers tend to write their code to fill up and use all available machine resources. Thus, as the capacity and speed of computers grow, the size and inefficiency of the code written for them seems to grow, too.

QAQA

Destructive Removal

Marty, you've often suggested "destructive removal" as a means of cleanly getting the 68B09E out of a dead CoCo 3. I want to add a little detail to your instructions: It is important to be careful to cut all the pins of the chip you are destructively removing very close to the body of the chip. This leaves more of the pin sticking up from the board, making it easier to grab with needle-nosed pliers when removing the pins one-by-one during the desoldering phase of the operation.

Lonnie McClure (LMCCLURE)
Little Rock, Arkansas

Thanks for the tip, Lonnie.

Martin H. Goodman, M.D., a physician trained in anesthesiology, is a longtime electronics tinkerer and outspoken commentator — sort of the Howard Cosell of the CoCo world. On Delphi, Marty is the SIGop of THE RAINBOW's CoCo SIG. His non-computer passions include running, mountaineering and outdoor photography. Marty lives in San Pablo, California.

OS-9 HOTLINE

GREG LAW

MV-Shell

I have a question concerning a program which was written by Dale Purkett and appeared in the June and July 1988 issues of THE RAINBOW. *The program in question is* MV-Shell, *which runs under* Multi-Vue. *I have the program and I have put together an AIF file. When I click on the icon, the program loads and, for a second, I get a menu. But then the whole thing erases itself and returns to the Tandy menu. Both the program and icon have the attributes set for owner execute and public execute. Do you have any ideas why this happens? One of the things I have tried is to load* `gfx2`, `syscall` *and* `inkey` *in memory ahead of time, but with no results. A copy of my AIF file is included.*

Finally, there are companion programs in the November 1988 issue called DoMenu *and* DoAlert. *Do these programs need their own AIF files to run? If not, how do I incorporate these programs into the* MV-Shell *module?*

Ernest Bazzinotti, Jr.
Dorchester, Massachusetts

I'm not sure why *MV-Shell* is aborting. It could be related to memory, or perhaps an error is being returned from the windowing system for some reason. You may want to load and run the original source from BASIC09 to determine if the program is running correctly. If it is, you may need to merge it with `inkey`, `gfx2`, and `syscall`. To do this, go into the `CMDS` directory and issue these commands:

```
rename mvshell.hak
merge mvshell.bak gfx2 syscall i
nkey >mvshell
attr mvshell e pe
```

This should considerably reduce the amount of overhead involved with loading each module individually.

All applications that run under *Multi-Vue* require an AIF file. In the case of *DoMenu* and *DoAlert*, you can use the same settings as you used for *MV-Shell*. That is, copy `aif.mvshell` to `aif.domenu` and `aif.doalert`, and change the application names from `mvshell` to `domenu` and `doalert`.

Where's OS-9?

When I ordered The Complete Rainbow Guide to OS-9 *and* The Complete Rainbow Guide to OS-9 Level II, Volume 1: A Beginner's Guide to Windows, *you sort of left me hanging. No where in the advertisement was there any mention of an OS-9 system master and I have never seen it advertised in* THE RAINBOW. *In fact, I have never seen it in any Radio Shack store or even heard of it until I got to Page 54 of* The Complete Rainbow Guide to OS-9 Level II. *I would also like to know if the OS-9 system master and* OS-9 Level II Operating System *are one and the same?*

Robert Cabral
USNS Kilauea

The OS-9 Level II operating system is currently available through Radio Shack Express Order (800-321-3133), although it used to be carried in the stores. The OS-9 Level II system master refers to the master disks included in the package. I'm not certain, but I believe the current price is $69.95. You also need at least a CoCo 3 and one disk drive, although 512K and two disk drives are highly recommended. As a matter of fact, I don't recommend using OS-9 without the 512K upgrade installed due to the extreme memory limitations in a 128K system.

Auto Won't Format

Enclosed is a copy of my OS-9 boot disk with the Auto Format *program on Page 72 of the March 1991 issue. I tried using* `/d1` *when the program asked which drive, but I still received Error 221. Here are the steps I've taken so far:*

```
copy /d1/cmds/auto /d0/cmds/auto
attr /d0/cmds/auto
```

From my boot disk, I type:

```
load auto
auto
```

The program asks to press a key, the disk name, number of disks, starting disk number and drive number. This is what is on my screen:

```
Formatting disk number 1 as #1
40 tracks
2 sides
You have error 221 in Auto Forma
tter
Continue (Y/N)?
```

I could use some help getting this program to work. Please see if you can find what I've done wrong.

L.T. Day
Zanesville, Ohio

I used the version of `auto` included on your disk, and it worked fine. I also compared the version of `auto` on your disk with my master and confirmed the packed files are the same. But it just occurred to me at the last minute that you probably do not have the `/nil` driver (`nildrv.dr` and `nil.dd`) since those files are included with *OS-9 Development System*. It can be a pain trying to remember which files are included with OS-9 Level II and which are included with *Multi-Vue* and *OS-9 Development System*. This makes sense, too, because `>/nil` is the only statement in the listing that would cause an Error 221 (module not found). All of the other statements would cause Error 216 (file not found) if you were missing an executable program such as `tmode` or `display`. All these little nuances can drive you batty sometimes. In short, you can fix the problem by either installing `nildrv.dr` and `nil.dd` in your `OS9Boot` file or by removing `>/nil` from the line that runs `format`.

Missing Menus

I was recently going through my RAINBOW ON DISK *library and found the source for* `locate`. *Wanting to use this enhancement to* `find`, *I entered the listings for* `gfx3` *and* `doalert`. *I then loaded all three modules and packed them. The problem I'm finding is that* `doalert` *will not create the window with menus. I've tried all types of screens, compared the source code for* `locate`, `doalert` *and* `gfx3`. *They all match the source as published in* THE RAINBOW. *The program just sits there after creating the arrow graphics cursor. I'm wondering if I missed a patch published at a later date?*

John Gilbertson
Portsmouth, Virginia

Although it wasn't specifically mentioned in the article, you need the `windint` module from the *Multi-Vue* disk in your `OS9Boot` file in order for `locate` to work properly. This module replaces `grfint` in the standard `OS9Boot` file and adds support for menu bars, the auto-follow mouse, etc. If you do not have *Multi-Vue*, you can order it from Radio Shack Express Order at (800) 321-3133.

UCSD Pascal

I am the owner of a CoCo 2 computer and have recently begun to study Pascal as implemented on the Apple computers we have at school. This has become a hassle. I recall seeing in your magazine some years ago an advertisement for a Pascal compiler for the CoCo 2. Do you have any information on software houses that would carry an implementation of UCSD Pascal that I could use on a 64K CoCo 2?

Donald Thomas
Dresden, Ohio

You are thinking of *DEFT Pascal* from DEFT Systems. Unfortunately this company is no longer in business and its products are no longer available. The only other Pascal compiler that might be available is *OS-9 Pascal*, which follows the ISO standard instead of the UCSD standard. You'll probably have to order *OS-9 Pascal* through Radio Shack Express Order.

In addition to being OS9 Online SIGop, Greg Law enjoys programming on all types of computers and has worked on systems ranging from the CoCo to the Burroughs B6700 super mainframe. He lives in Louisville, Kentucky.

Submitting Material To Rainbow

Contributions to THE RAINBOW are welcome from everyone. We like to run a variety of programs that are useful, helpful and fun for other CoCo owners.

WHAT TO WRITE: We are interested in what you want to tell our readers. We accept for consideration anything that is well-written and has a practical application for the Tandy Color Computer. If it interests you, it will probably interest lots of others. However, we vastly prefer articles with accompanying programs that can be entered and run. The more unique the idea, the more the appeal. We have a continuing need for short articles with short listings. These are especially appealing to our many beginners.

FORMAT: Program submissions must be on tape or disk, and it is best to make several saves, at least one of them in ASCII format. We're sorry, but we do not have time to key in programs and debug our typing errors. All programs should be supported by some editorial commentary explaining how the program works. We also prefer that editorial copy be included in ASCII format on the tape or disk, using any of the word processors currently available for the Color Computer. Also, please include a double-spaced printout of your editorial material and program listing. Do not send text in all capital letters: use upper- and lowercase.

COMPENSATION: We do pay for submissions, based on a number of criteria. Those wishing remuneration should *so state* when making submissions.

For the benefit of those wanting more detailed information on making submissions, please send a self-addressed, stamped envelope (SASE) to: Submission Guidelines, THE RAINBOW, The Falsoft Building, P.O. Box 385, Prospect, KY 40059. We will send you comprehensive guidelines.

Please do not submit material currently submitted to another publication.

Call for Printer Software

The CoCo is a great little computer even without all the add-ons. But let's face it, the add-ons (disk drives, modems, etc.) make computing life even easier. The printer has long been one of the first peripheral devices we'll recommend to users wanting to upgrade their systems. And with good reason: Viewing screen output is OK, so long as someone else doesn't need a copy. If you've written a program for using a printer with the CoCo, perhaps someone else could use it, too.

We are now making tentative plans for the May 1993 issue of THE RAINBOW and are accepting program submissions appropriate for that issue's theme, Printers. We are also interested in general-interest articles discussing how printers can be used with the CoCo. All submissions must be received by us no later than January 29, 1992, and must follow our standard submission guidelines (see Page 15 for details and address).

We'd also like to see any other programs or articles you have written (submitted material must be the original work of the submitting party, or submitted with written permission). All submissions are evaluated and considered for publication in future issues.

About Your Subscription

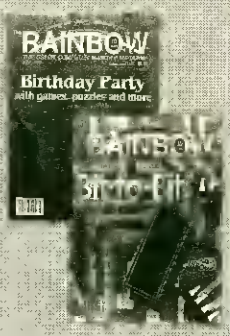
Your copy of THE RAINBOW is sent second class mail. You must notify us of a new address when you move. Notification should reach us no later than the 15th of the month prior to the month in which you change your address. Sorry, we cannot be responsible for sending another copy when you fail to notify us.

Your mailing label also shows an account number and the subscription expiration date. Please indicate this account number when renewing or corresponding with us. It will help us help you better and faster.

For Canadian and other non-U.S. subscribers, there may be a mailing address shown that is different from our editorial office address. Send your correspondence to our editorial offices at Falsoft, Inc., The Falsoft Building, P.O. Box 385, Prospect, KY 40059.

Yes! They're still available!

RAINBOW Back Issues



BACK ISSUES STILL AVAILABLE
Have you explored the wealth of information in our past issues? From our very first, four-page issue to many with more than 300 pages of material, it's all just for CoCo users — a great way to expand your library!

A WORLD OF INFO AT A BARGAIN PRICE
All back issues sell for the single issue cover price. In addition, there is a $3.50 charge for the first issue, plus 50 cents for each additional issue for postage and handling if sent by United Parcel Service. There is a $5 charge for the first issue, plus a $1 charge for each additional issue on orders sent by U.S. Mail. UPS will not deliver to a post office box or to another country.

MOST ISSUES STILL AVAILABLE
Available issues through June 1982 are provided on white paper in a reprint form. All others are in regular magazine form. VISA, MasterCard and American Express accepted. Kentucky residents please add 6 percent sales tax; Canadian residents, 7 percent GST. In order to hold down costs, we do not bill, and no C.O.D. orders are accepted.

Due to heavy demand, we suggest you order the back issues you want now while supplies last.
To order, review and fill out the form below and mail it with your payment.

For greater convenience, order through the Rainbow Magazine Services area of our Delphi CoCo SIG.

RAINBOW INDEX
A complete index for July 1981 through June 1984, is printed in the July 1984 issue. Separate copies are available for $2.50 plus 50¢ handling. Indexes for subsequent years are published annually in the July issues of THE RAINBOW.

TOTAL _____
KY RESIDENTS ADD 6% _____
CANADIAN RESIDENTS ADD 7% GST _____
U.S. MAIL CHARGE _____
SHIPPING & HANDLING _____
U.P.S. CHARGE _____
TOTAL AMOUNT _____
ENCLOSED _____

Article Reprints
In instances where a given issue is now out of print and not available for purchase, we do provide photocopies of specific articles. The cost for this service is $1.50 plus 50 cents S/H per article. This service is provided *only* in the case of out-of-stock issues.

Name _____
Address _____
City _____ State _____ Zip _____
❑ Payment Enclosed, or Charge to my: ❑ VISA ❑ MC ❑ AE
Card # _____
Expiration Date _____ Phone ( ) _____
Signature _____

TO ORDER BY PHONE (credit card orders only) call (800) 847-0309, 9 a.m. to 5 p.m. EST. All other inquiries call (502) 228-4492.
send to: THE RAINBOW, The Falsoft Building, P.O. Box 385, Prospect, KY 40059

Please send me the following back issues:

	VOLUME 1		APR 84	Gaming	$3.95 ❑	DEC 84	Holiday	$3.95 ❑	FEB 89	Home Help	$3.95 ❑
JUL 81	Premier Issue	$2.00 ❑	MAY 84	Printer	$3.95 ❑	JAN 85	Beginners	$3.95 ❑	MAR 89	Hardware	$3.95 ❑
FEB 82		$2.00 ❑	JUN 84	Music	$3.95 ❑	FEB 85	Utilities	$3.95 ❑	APR 89	Business	$3.95 ❑
	VOLUME 2		JUL 84	Anniversary	$3.95 ❑	MAR 85	Business	$3.95 ❑	MAY 89	Printer	$3.95 ❑
JUN 83	Printers	$2.95 ❑				APR 85	Simulations	$3.95 ❑	JUN 89	Summer Fun	$3.95 ❑
	VOLUME 3			VOLUME 4		MAY 85	Printer	$3.95 ❑	JUL 89	Anniversary	$3.95 ❑
AUG 83	Games	$2.95 ❑	AUG 84	Games	$3.95 ❑	JUN 85	Music	$3.95 ❑			
SEP 83	Education	$2.95 ❑	SEP 84	Education	$3.95 ❑	JUL 85	Anniversary	$3.95 ❑		VOLUME 9	
OCT 83	Graphics	$3.95 ❑	OCT 84	Graphics	$3.95 ❑				AUG 89	Beyond BASIC	$3.95 ❑
MAR 84	Business	$3.95 ❑	NOV 84	Data Comm.	$3.95 ❑		VOLUME 5		SEP 89	Education	$3.95 ❑
						AUG 85	Games	$3.95 ❑	OCT 89	Graphics	$3.95 ❑
						SEP 85	Education	$3.95 ❑	NOV 89	Data Comm.	$3.95 ❑
						OCT 85	Graphics	$3.95 ❑	DEC 89	Holiday	$3.95 ❑
						NOV 85	Data Comm.	$3.95 ❑	JAN 90	Beginners	$3.95 ❑
						DEC 85	Holiday	$3.95 ❑	FEB 90	Home Help	$3.95 ❑
						JAN 86	Beginners	$3.95 ❑	MAR 90	Hardware	$3.95 ❑
						FEB 86	Utilities	$3.95 ❑	APR 90	Business	$3.95 ❑
						MAR 86	Business	$3.95 ❑	MAY 90	Printer	$3.95 ❑
						APR 86	Home Help	$3.95 ❑	JUN 90	Summer Fun	$3.95 ❑
						MAY 86	Printer	$3.95 ❑	JUL 90	Anniversary	$3.95 ❑
						JUN 86	Music	$3.95 ❑			
						JUL 86	Anniversary	$3.95 ❑		VOLUME 10	
									AUG 90	OS-9	$3.95 ❑
							VOLUME 6		SEP 90	Education	$3.95 ❑
						AUG 86	Games	$3.95 ❑	OCT 90	Graphics	$3.95 ❑
						SEP 86	Education	$3.95 ❑	NOV 90	Data Comm.	$3.95 ❑
						OCT 86	Graphics	$3.95 ❑	DEC 90	Holiday	$3.95 ❑
						NOV 86	Data Comm.	$3.95 ❑	JAN 91	Beginners	$3.95 ❑
						DEC 86	Holiday	$3.95 ❑	FEB 91	Home Help	$3.95 ❑
						JAN 87	Beginners	$3.95 ❑	MAR 91	Hardware	$3.95 ❑
						FEB 87	Utilities	$3.95 ❑	APR 91	Music	$3.95 ❑
						MAR 87	Business	$3.95 ❑	MAY 91	Printer	$3.95 ❑
						APR 87	Home Help	$3.95 ❑	JUN 91	Summer Fun	$3.95 ❑
						MAY 87	Printer	$3.95 ❑	JUL 91	Anniversary	$3.95 ❑
						JUN 87	Music	$3.95 ❑			
						JUL 87	Anniversary	$3.95 ❑		VOLUME 11	
									AUG 91	Graphics	$3.95 ❑
							VOLUME 7		SEP 91	Education	$3.95 ❑
						AUG 87	Games	$3.95 ❑	OCT 91	OS-9	$3.95 ❑
						SEP 87	Education	$3.95 ❑	NOV 91	Data Comm.	$3.95 ❑
						OCT 87	Graphics	$3.95 ❑	DEC 91	Holiday	$3.95 ❑
						NOV 87	Data Comm.	$3.95 ❑	JAN 92	Utilities	$3.95 ❑
						DEC 87	Holiday	$3.95 ❑	FEB 92	Home Help	$3.95 ❑
						JAN 88	Beginners	$3.95 ❑	MAR 92	Hardware	$3.95 ❑
						FEB 88	Utilities	$3.95 ❑	APR 92	Music	$3.95 ❑
						MAR 88	Business	$3.95 ❑	MAY 92	Printer	$3.95 ❑
						APR 88	Home Help	$3.95 ❑	JUN 92	Programming	$3.95 ❑
						MAY 88	Printer	$3.95 ❑	JUL 92	Anniversary	$3.95 ❑
						JUN 88	Music	$3.95 ❑			
						JUL 88	Anniversary	$3.95 ❑		VOLUME 12	
									AUG 92	Graphics	$3.95 ❑
							VOLUME 8		SEP 92	Education	$3.95 ❑
						AUG 88	Games	$3.95 ❑	OCT 92	OS-9	$3.95 ❑
						SEP 88	Education	$3.95 ❑	NOV 92	Data Comm.	$3.95 ❑
						OCT 88	Graphics	$3.95 ❑	DEC 92	Holiday	$3.95 ❑
						NOV 88	Data Comm.	$3.95 ❑	JAN 93	Utilities	$3.95 ❑
						DEC 88	Holiday	$3.95 ❑			
						JAN 89	Beginners	$3.95 ❑			

Code from cover

printer.code. For *Sendcode* to work properly, all .code files must be in the /dd/SYS directory on the system. A sample .code file for the DMP-132 printer is shown in Figure 1. If you have a DMP-132, go ahead and create this file by using the OS-9 build command or a text editor, then you can start using *Sendcode* right away. If you need or want to devise a different file, read on.

```
/p
bell 0 7
LineFeed 0 10
FormFeed 0 12
CR 0 13
UndrlinON  0 15
UndrlinOFF 0 14
GraphixON  0 18
GraphixOFF 0 30
WordProc 0 20
ReverseLF 0 27 10
1_8LF 0 27 26
1_2LF 0 27 28
1_12LF 0 27 50
1_36LF 0 27 51
3_4LF 0 27 56
1_144LF 0 27 57
1_144LFs 1 27 64
ElongatON  0 27 14
ElongatOFF 0 27 15
NLQprop 0 27 17
NLQpica 0 27 18
Pica 0 27 19
Compressed 0 27 20
CRonly 0 27 21
CRandLF 0 27 22
Elite 0 27 23
BoldON  0 27 31
BoldOFF 0 27 32
IBMmode 0 27 33
PageLength 1 27 52
IBMSet2  0 27 58
TandySet 0 27 59
ItalicsON  0 27 66 1
ItalicsOFF 0 27 66 0
PerfSkip 1 27 72
MicroFont 0 27 77
LeftMargin 1 27 81
RghtMargin 1 27 82
SuperON 0 27 83 0
SubON   0 27 83 1
BiDirect  0 27 85 0
UniDirect 0 27 85 1
SuperOFF 0 27 88
SubOFF   0 27 88
Country 1 27 89
Repeat 2 28
```

Figure 1: DMP-132 printer.code ***File***

The first line of the .code file specifies the device or path where you want the control codes sent. Typically this would be /p for your printer. You can also specify stdout or stderr if you want the codes sent to the standard output or standard error path (more on this in a moment).

Your command definitions appear on the succeeding lines. The first item that appears on each line is the command name you want to use. This command name can be up to 10 characters in length and may use upper- and lowercase. However, remember that *Sendcode*'s command search is not case-sensitive — no distinction is made between upper- and lowercase letters.

The first number following the command name on each line tells *Sendcode* how many user-supplied codes are required for that command. *Sendcode* allows up to three user-provided codes. We'll examine this feature more closely in a moment.

The remaining numbers on each line are the control codes to be sent for the command name on that line. These numbers are in decimal format and can range from 0 to 255. *Sendcode* supports up to five control-code values for each command you define.

The last character in each line must be a carriage return (ENTER). When building a .code file, remember that each line can contain only one command. You can define as many commands as you like, but each command, along with its control codes, must be on its own line.

When you execute *Sendcode*, you can enter up to 30 defined command names on the command line. In other words, you can ask *Sendcode* to send codes for up to 30 separate functions in one OS-9 command line. This should be more than enough for most uses.

Now let's take a look at user-supplied codes. There are probably a few control codes you won't want to predefine. For example, many printers allow you to set the left margin at any character position. It would be horrible to have to define 80 different command names so you could set the margin at any position. Instead, *Sendcode* allows you to send the character position as a parameter on the command line. To do this, you enter the defined command name along with the value you want to send, enclosing the value in parenthesis. For example, if the command LeftMargin is defined in printer.code and specifies a user-supplied codes value of 1 (see Figure 1), you would enter

```
sendcode LeftMargin(10)
```

to set the left margin to 10 character spaces. Note that there are no spaces between any of the characters in the command name/user values entered on the command line.

Since user-supplied values usually trail a defined sequence of control codes, they are sent after any predefined codes are sent. The codes are sent starting with the left-most number and ending with the right-most number. For example, when you enter the above command line, *Sendcode* first finds the command definition in the .code file. It then determines that one user-supplied code is expected on the OS-9 command line. (In this case, the user-supplied value is 10.) Then *Sendcode* sends any codes specified for the command in the .code file (in this case, 27 followed by 81). After the defined codes are sent, *Sendcode* sends the user-supplied value of 10.

Up to three user-supplied codes may be defined for each command name you specify in the .code file. When executing a command that requires two or three user supplied codes, separate the values with dashes (-). The following is an example:

```
sendcode Repeat(10-32)
```

Again, there can be no spaces between any of the characters of the command name. (The reason I wrote *Sendcode* to use dashes instead of commas or spaces is to simplify the program. OS-9 parses each *parameter* on the command line by looking for spaces and commas. By using dashes, OS-9 does not split the command line into several parameters.)

As I mentioned before, the default .code file used by *Sendcode* is printer.code. If you want, you can change the name printer in the source code before compiling the program. (It is defined in the Global Variables section near the beginning. Each character of the name you use must be in single quotes, then separated by a comma.) However, don't change the defined path and extension since *Sendcode* uses these strings as defaults elsewhere.

Sendcode handles multiple devices easily. Suppose you have two printers that use different control codes. Simply enter the definitions for the printer you use most in the printer.code file (the default). Then build a .code file for the other printer, using a filename that identifies that printer. To tell *Sendcode* to use the second .code file, enter the name of that file, preceded by a dash, as a parameter on the OS-9 command line. For example, if your second printer is a Star NX-1000 and you name its .code file nx1000.code, you might enter

```
sendcode -nx1000 UndrlnON
```

to turn its underlining feature on. This example assumes there is a file called nx1000.code in the /dd/SYS directory (remember, all .code files must be in this directory) and that an UndrlnON command has been defined in that file.

By using the device names stdout and stderr, you can also send control codes to the screen. Figure 2 shows a listing of a .code file that includes screen functions. Notice that the device specified on the first line is stdout. To use this file to ring the bell, you would enter

```
sendcode -screen Bell
```

```
stdout
Home 0 1
Cursor 2 2
CursorOFF 0 5 32
CursorON 0 5 33
Bell 0 7
EraseEOS 0 11
CR 0 13
ReverseON 0 31 32
ReverseOFF 0 31 33
UndrlnON 0 31 34
UndrlnOFF 0 31 35
BlinkON 0 31 36
BlinkOFF 0 31 37
```

Figure 2: Sample screen.code ***File***

Feel free to change the command definitions in screen.code to support the functions you want. Users of OS-9 Level II should find this approach very useful for handling windowing functions.

In case you forget the commands you defined in the .code file, enter a question mark (?) instead of a command definition. If you enter

```
sendcode ?
```

the definitions in the .code file are displayed onscreen. This is handy for checking suspect .code files; if the information is displayed incorrectly, you have typed something incorrectly in the .code file. To list the command definitions for a different .code file, add the filename to the command line, as in the following example:

```
sendcode -nx1000 ?
```

Sendcode has certainly made my computing more enjoyable and productive. I hope you find it to be a handy utility, too.

Bruce Geren is a computer engineer for Motorola. He and his wife, Launa, have two children, Alan and Megan, and another child on the way. Bruce may be contacted at 1586 W. Maggio Way, Apt. 2113, Chandler, AZ 85224. Please include an SASE when requesting a reply.

OS-9

The Listing: Sendcode.c

```
/* sendcode.c
 *
 * Copyright (c) 1990 by Bruce Geren
 *
 */

#include <stdio.h>

/* global definitions */
#define TRUE   1
#define FALSE  0

/* type definitions */
typedef int void;
typedef int boolean;
typedef struct codes {
  char command[12];
  int numcodes,
      code1,
      code2,
      code3,
      code4,
      code5;
  int numextra,
      xcode1,
      xcode2,
      xcode3;
  } CODETYPE;

/* forward referencing of local functions */
void strtolower();
void dumpcodes();

/* global variables */
char codefn[81] = {'/','d','d','/','s','y','s','/',
                   'p','r','i','n','t','e','r','.','c','o','d','e'};
CODETYPE init_tc = {{'\0'},0,0,0,0,0,0,0,0,0,0};

char *usage1 = "\nsendcode [-peripheral] [code[code[...]]]";
char *usage2 = "  where code = control_code [{code1[-code2[-code3]]}]\n";
char *nonesent = "No codes sent due to error(s)";
char *codefrmt = "%10s %d %d %d %d %d %d\n";


int main(argc, argv)
int argc;
char *argv[];
  {
  int firstparam = 1;
  register int i;
  int cmnd_err;
  int *codeptr;
  boolean stdio = FALSE;
  char devicename[20];
  char in_str[81];
  char *chptr;
  FILE *fp;
  FILE *codefp;
  CODETYPE lc[30];
  CODETYPE tc;

  /* display usage if no parameters */
  if (argc == 1) {
    puts(usage1);
```

```
    puts(usage2);
    exit(0);
    }
  argc--;

  /* check for change of code file name */
  if (*argv[1] == '-') {
    strcpy(codefn,"/dd/sys/");
    strcat(codefn,(argv[1]+1));
    strcat(codefn,".code");
    firstparam = 2;
    argc--;
    }

  /* check for code file content listing request */
  if (*argv[firstparam] == '?') {
    puts(usage1);
    puts(usage2);
    dumpcodes();
    exit(0);
    }

  /* create a list of commands from the parameter list */
  for (i = 0; i < argc; i++) {
    _strass(&lc[i], &init_lc, sizeof(CODETYPE));
    strncpy(lc[i].command, argv[i + firstparam], 11);
    lc[i].command[11] = '\0';
    if ((chptr = index(lc[i].command,'(')) != NULL)
      *chptr = '\0';
    strtolower(lc[i].command);
    }

  /* open code file for command interpreting */
  if ((fp = fopen(codefn,"r")) == NULL)
    exit(errno);

  /* read device name and special case stdout and stderr */
  fscanf(fp, "%s\n",devicename);
  strtolower(devicename);
  if (strcmp(devicename,"stdout") == 0) {
    codefp = stdout;
    stdio = TRUE;
    }
  else if (strcmp(devicename,"stderr") == 0) {
    codefp = stderr;
    stdio = TRUE;
    }

  /* read each command with its control codes */
  while (fgets(in_str,80,fp) != NULL) {
    tc.numcodes = sscanf(in_str, codefrmt, tc.command, &tc.numextra,
      &tc.code1, &tc.code2, &tc.code3, &tc.code4, &tc.code5) - 2;

    strtolower(tc.command);
    for (i = 0; i < argc; i++)
      if (strcmp(tc.command, lc[i].command) == 0)
        _strass(&lc[i], &tc, sizeof(CODETYPE));
    } /* while */

  if (ferror(fp))
    exit(fp);

  /* look for any parameters that had */
  /* no commands in the code file     */
  cmnd_err = FALSE;
  for (i = 0; i < argc; i++)
    if (lc[i].numcodes == 0) {
      printf("Unknown command --> %s\n",lc[i].command);
      cmnd_err = TRUE;
      }

  if (cmnd_err) {
    puts(nonesent);
    exit(0);
    }

  /* check for user provided codes */
  cmnd_err = FALSE;
  for (i = 0; i < argc; i++) {
    chptr = index(argv[i + firstparam], '(');

    /* display error for unwanted user provided codes */
    if ((lc[i].numextra == 0) && (chptr != NULL)) {
      printf("Extra code(s) not required --> %s\n",
              argv[i + firstparam]);
      cmnd_err = TRUE;
      }
    else if (lc[i].numextra > 0) {
      /* display error if missing user required codes */
      if (chptr == NULL) {
        printf("Missing user provided codes --> %s\n",
                argv[i + firstparam]);
        cmnd_err = TRUE;
        }

      /* else display error if wrong # of user required codes */
      else {
        if (sscanf(chptr,"(%d-%d-%d",
            &lc[i].xcode1, &lc[i].xcode2, &lc[i].xcode3)
                != lc[i].numextra) {
          printf("Wrong number of user provided codes --> %s\n",
                  argv[i + firstparam]);
          cmnd_err = TRUE;
          } /* else */
        } /* if > 0 */
      } /* for */

    if (cmnd_err) {
      puts(nonesent);
      exit(0);
      }

    /* send control codes to the device */
    if (!stdio)
      if ((codefp = fopen(devicename,"w")) == NULL)
        exit(errno);

    /* for each command ... */
    for (i = 0; i < argc; i++) {

      /* send normal control code(s) */
      codeptr = &lc[i].code1;
      while (lc[i].numcodes--)
        putc(*codeptr++, codefp);

      /* send user required control code(s) */
      codeptr = &lc[i].xcode1;
      while (lc[i].numextra--)
        putc(*codeptr++, codefp);
      } /* for */

    if (ferror(codefp))
      exit(errno);

    if (!stdio)
      fclose(fp);
    } /* main - sendcode */


  /* convert entire string to lower case */
  void strtolower(str)
  char *str;
    {
    while (*str++ = tolower(*str));
    }


  /* dump control codes to standard output for reference */
  void dumpcodes()
    {
    int num_codes;
    register int i;
    int *codeptr;
    FILE *fp;
    CODETYPE tc;
    char device[21];
    char in_str[81];
    char tempfn[81];
    char *chptr;

    if ((fp = fopen(codefn,"r")) == NULL)
      exit(errno);

    /* print usage and heading information */
    fscanf(fp,"%20s\n",device);
    strcpy(tempfn, codefn);
    chptr = rindex(tempfn,'.');
    *chptr = '\0';
    chptr = 1 + rindex(tempfn,'/');
    printf("Control Codes for peripheral \"%s\"\n",chptr);
    printf("using device \"%s\"\n\n",device);
    puts("          Codes:                  # Extra");
    puts(" Command    1   2   3   4   5  Codes");
    puts("-----------|--- --- --- --- ---|-  ---");

    /* read and display each command with its control codes */
    while (fgets(in_str,80,fp) != NULL) {
      num_codes = sscanf(in_str, codefrmt, tc.command, &tc.numextra,
        &tc.code1, &tc.code2, &tc.code3, &tc.code4, &tc.code5) - 2;
      printf("%-10s", tc.command);
      codeptr = &tc.code1;

      for (i = 5; i > 0; i--, num_codes--) {
        if (num_codes > 0)
          printf(" %3d",*codeptr++);
        else
          printf("    ");
        }

      printf(" %4d\n", tc.numextra);
      }

    if (ferror(fp))
      exit(errno);

    fclose(fp);
    }
```

Sundog Systems recently announced it has completed a project started over two years ago. Led by Jeff Steidl, author of *Photon* and *GrafExpress 2.0*, three programmers combined their efforts to produce *Contras*. This 512K CoCo 3 program features full-screen (320-by-225) hardware scrolling, smooth animation, background music, sound effects and arcade action. It also supports both one- and two-player modes.

According to Glen Dahlgren of Sundog Systems, "Jeff proved that the CoCo can match — or surpass — any home game system. This is paramilitary combat at its best." While we haven't received our copy yet, you can bet this is one new CoCo product we'll have a blast with. Look for the upcoming review in THE RAINBOW.

Feature Program

Clean the Screen

by Steven Puls

Are you bored with the way Color BASIC's CLS command works? Do you wish you had a more interesting way to clear the CoCo's 32-column screen? If, so, *NewCLS* could be the answer for you.

NewCLS is a short utility that adds a little spice to the way the CoCo clears its standard screen. Best of all, *NewCLS* works on any CoCo with at least 16K of memory.

To use this utility, enter the program shown in the listing and save it to tape or disk. This BASIC program stores in memory a machine-language routine that handles the actual work of clearing the screen. It then saves this routine to disk. (Readers with tape-based CoCo systems should change SAVEM in Line 10 to CSAVEM. Also make sure you press the Record and Play buttons on the tape recorder before you run the BASIC program.)

To execute the machine-language routine created by *NewCLS*, you must first load it into memory. To do this, enter CLEAR 100,&H3000 followed by LOADM"NEWCLS". (Tape users should enter CLOADM"NEWCLS".)

Once the routine is in memory, simply enter EXEC to clear the screen. Alternatively, *NewCLS* can be used by your other BASIC programs; just load and execute it by issuing the above commands under program control.

You can change the screen colors and patterns by entering POKE &H300A,*x*, where *x* is any value between 0 and 255, before executing the program. Experiment and see what values work best for you. I hope you find *NewCLS* to be a useful little program.

Steven Puls is currently a junior in high school. Since he received his first CoCo six years ago, he has enjoyed writing programs for it. Steven hopes to make a career of computer programming.

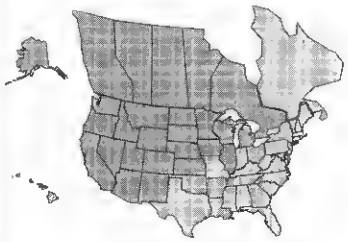
16K

The Listing: NEWCLS

```
1 'NEW CLS
2 'BY STEVEN PULS
3 'COPYRIGHT (C) 1992
4 'BY FALSOFT, INC.
5 'RAINBOW MAGAZINE
9 CLEAR 100,&H3000:GOTO30
10 SAVEM"NEWCLS",&H3000,&H3029,&
H3000
20 END
30 FORADD=&H3000 TO&H3029:READIN
F$:POKEADD,VAL("&H"+INF$):NEXT:G
OTO 10
40 DATA 8E,4,0,10,8E,0,0,A6,84,8
1,20,27,6,80,1,A7,84,31,21,30,1,
8C,6,0,26,ED,10,8C,0,0,27,9,10,8
E,0,0,8E,4,0,20,DE,39
```

Advertisers Index

We appreciate your mentioning THE RAINBOW when you contact these advertisers.



Call Kim Lewis
Eastern Sales Director
(502)228-4492

Call Ira Barsky
Western Sales Director
(312)587-1818

The Falsoft Building
9509 U.S. Highway 42, P.O. Box 385, Prospect, KY 40059
(502) 228-4492 • FAX (502) 228-5121

SUNDOG SYSTEMS

PHOTON

NEW!

Energy is everything, your home world depends on it. However someone or something is slowly siphoning it away. As your world's champion, you must climb into the experimental Power Tank to challenge this nemesis and his minions. Your key lies with the ability to teleport solid mass. Use this to manipulate and explore the endless stronghold of the enemy, and to exploit the free-floating DUPES (Dense Units of Photon Energy) to destroy the menacing Plasma Droids. Be cautious, though: those DUPES can be deadly, too! Photon, a fantastic new arcade game for your Coco3, contains spectacular 320x200 resolution, 16 color graphics, ultra-smooth 60 Hz animation, and loads of real-time music and sound effects. It will send your mind racing over endless possibilities, requiring quick decisions and reactions. Quite simply, Photon is incredibly addictive; it will deliver hours of excitement. Will you become your world's greatest hero, or just another energy slave? Req. 128K CoCo 3 and disk drive.

$34.95

GRAF 2.0 EXPRESS

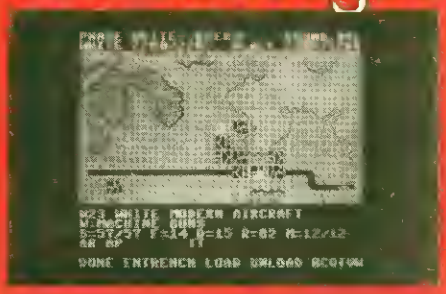
NEW!

GrafExpress 2.0 is a complete graphics and music programming environment. From the beginner to the accomplished professional, you can use GrafExpress to create lightning fast arcade games, graphic applications and utilities, and windowing multimedia demonstrations! The GrafExpress package includes two incredible systems. GrafExpress 16 works on all monitor types and offers support in 12 graphic resolutions (from 128x192 to 320x225). GrafExpress 256 offers 6 resolutions (from 128x192 to 160x225 on a composite monitor in an astounding 256 colors! Ever see a CoCo do that before? Both systems include standard graphics commands (CIRCLE, FILL, etc.) that blow away the competition. For example, the BOX command peaks out at over 2 MegaPixels/second; that's 300 times faster than BASIC! 255 separate sprites of up to 100x100 pixels each are supported with window clipping and high-res pixel level collision checking. The 8-octave/4-voice music synthesizer has independent envelope, waveform, and volume controls, a 7+ KHz sampling rate, and much more. Other features include text/graphics mixing, different font sizes, fast window copying and scrolling, picture save/load, easy implementation from both BASIC and assembly language, multiple screen animation, and support for 128K/512K, double speed, and the high-res joystick interface. The package also contains support programs that are worth the purchase price of GrafExpress alone! These include an introductory demo, a picture editor, a waveform editor, and an art program that supports 256 colors! GrafExpress also comes with a 50 page manual that fully explains all of its incredible features. If you do any graphics programming or simply want to see what your little CoCo is capable of, GrafExpress is a must! Req. 128K CoCo 3 and disk drive.

$34.95

War Monger

NEW!



The world is in unrest. Power hungry villains and evil warlords are readying their forces. It falls to you to lead your people against these armies, and only your best strategic plans can save the day. Fight the good fight in any era or locale. Play a simple game of capture the flag armed with water balloons, or climb into the cockpit of a 100 foot high armored warrior. Explore the deepest dungeons, defend your galaxy, or create your own scenarios with this incredible war game construction set/simulator. Your imagination is your only limit. You will deploy your forces with total control over hostile terrain while you scroll a graphic bird's-eye window over an immense world. War Monger has terrific 320x200 resolution, 16 color graphics and includes a tile editor to create or edit your own. Play against the computer, battle with another player, or simply watch the computer plot against itself. The enemy is everywhere. Are you ready to take on the challenge as the War Monger? Req. 128K CoCo 3 and disk drive.

$29.95

The Quest for THELDA

An immensely popular 128K CoCo 3 arcade/adventure. Over 500 screens of fast fantasy action and puzzle solving. Great graphics and sound effects. **$34.95**. Hint book only $4.95.

★ THE CONTRAS ★

A 512K two player futuristic combat arcade game. Full screen 320x225 hardware scrolling and smooth animation. Background music score and sound effects! 512K Coco3 only. **34.95**. Shipping soon!

Crystal City

This was THE game of '91! Ultra-fast space action with hardware scrolling on a 128K CoCo 3. Wild sound effects and over 30 MegaBytes of amazing graphics! **34.95**.

ZENIX

Lightning fast arcade game for the 128K CoCo 3. Terrific 320x225 graphics, background music score and sound effects, and out-of-sight game play. **$29.95**.

ALSO AVAILABLE

Warrior King CoCo 3	$29.95
In Quest of the Star Lord CoCo 3	$34.95
Hint Sheet	$ 3.95
Hall of the King 1, 2 or 3 CoCo 1-3	$29.95 ea.
Hall of the King Trilogy	$74.95
White Fire of Eternity CoCo 1-3	$19.95
Dragon Blade CoCo 1-3	$19.95
Champion CoCo 1-3	$19.95
Paladin's Legacy CoCo 1-3	$24.95

Visa, Mastercard, Check, Money Order, and COD (USA only, please) accepted. All foreign orders must be sent in US currency Money Orders. Include $2.50 for shipping in USA and Canada. $5.00 Foreign. $3.00 extra for COD orders. PA residents add 6% sales tax. Dealer inquiries welcome. Authors, we're looking for new software!

Sinistaar

Everyone loves this 512K arcade game. 3 disks packed with spectacular graphics and eerie background digital sound effects. 512K CoCo 3 only. **$34.95**.

KYUMI-SAI TO BE NINJA

The best selling 128K CoCo 3 martial arts arcade game. Now available in both RS-DOS and OS-9 versions. Play the incredible combat experience you've been missing under the operating system of your choice! **$29.95**.

SOUNDSMITH

A polyphonic digital sound sequencing system for your 128K/512K CoCo 3 with a user-friendly point-and-click graphic editor. Create music scores with your own sounds or from the many we provide. **$34.95**.

Sample instrument disks. 6 sides of sampled sounds/instruments. Only $12.95 each or $29.95 for all three.

Sundog systems

SUNDOG SYSTEMS

P.O. Box 766 • Manassas, VA 22111

703/330-8989